



**IM IN UR FLOWSPAZES ACTIONIN
UR ITEMZ**

“Like Twitter, for photos”

Alpha Alpha! Disco Ball!!

“Like Twitter, for photos”

Steve Jobs' comments about the iPhone's "**controlled environment**" <http://the.taoofmac.com.nyud.net:8080/space/blog/2007-01-14> " were probably the best thing the people at Nokia heard all week.

I occasionally get asked how I got in to computer programming and why I keep at it. The answer is pretty simple : Because no one else seems to want to write the kinds of applications that I want to use. This is even more true of mobile-computing-device-phones :

Sometime between 2007 and 2014 these photos were lost in translation.

There's still *a lot* of polish needed on this thing, but the basic framework is there to aggregate, trigger an alarm (read: make the phone vibrate) and display photos from any Flickr API method that returns the "standard photos response". It looks like this :

```
1 | # s60_include : /home/asc/lib/python/s60-simplejson/s60-simplejson.py
2 | # s60_include : /home/asc/lib/python/s60-simplehttp/s60-simplehttp.py
3 | # s60_include : /home/asc/lib/python/s60-simplestore/s60-simplestore.py
4 | # s60_include : /home/asc/lib/python/s60-simplenumbers/s60-simplenumbers.py
5 | # s60_include : /home/asc/lib/python/s60-simpleflickr/s60-simpleflickr.py
6 | # s60_include : /home/asc/lib/python/s60-simpleapp/s60-simpleapp.py
7 | # s60_include : /home/asc/lib/python/s60-simpleflickrapp/s60-simpleflickrapp.py
8 |
9 | class flickrwdgt (simpleflickrapp) :
10 |
11 |     def get_flickr_photos (self) :
12 |         return self.flickr_api_call("flickr.photos.getContactsPhotos" , {'count':50})
13 |
14 | if __name__ == '__main__' :
15 |
16 |     key    = 'yer flickr api key'
17 |     secret = 'yer flickr app secret'
18 |     token  = 'yer flickr auth token'
19 |
20 |     app = flickrwdgt(key, secret, token)
21 |     app.loop()
```

Which is convenient since the `flickr.photos.search` method could just as easily be dropped in and Python could do something crazy like ... talk to a GPS device and show nearby photos. I actually had a no-less finished version of this working for 2nd Edition Python so it ought to be easy enough to get it working for 3rd edition. Easy.

The part where you have to hard-code the Flickr API tokens is not a feature. I just haven't figured out *what* or *how* I'm going to do about it. If I can get at whatever folder SMS messages or web downloads are stored in then I suppose I will just try to read configs from there.

Until then, here's the un-SIS-ified source code if you've got the stomach to plug in your own values and bundle it up in to a binary : **s60-**

flickrwidgt_build_noauth.py `/weblog/2007/01/03/itemz/s60-flickrwidgt_build_noauth.py`

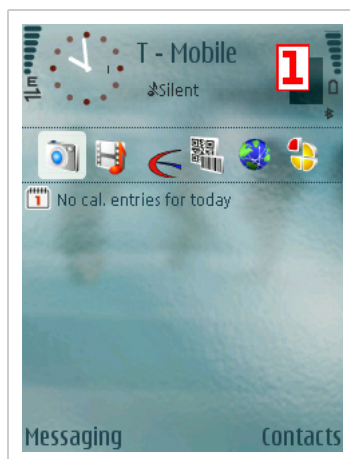
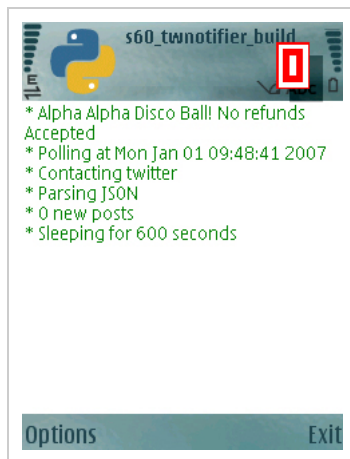
Finally, if I rebuild the **Twitter notification agent** `http://www.aaronland.info/weblog/2007/01/03/itemz/#disco` with all the **s60-*** libraries listed above and it Just Works™, then I will spend my morning coffee this week writing documentation and releasing proper packages.

2007-01-02T22:45:42-0800

Alpha Alpha! Disco Ball!!

At this point, I am just writing documentation away from releasing proper packages for the code that runs my new minty fresh Twitter notification agent (that actually works under Series 60 3rd Edition Python).

Aside from the part where it sits in the background and asks Twitter for updates every ten minutes, it will also make your phone vibrate when new messages arrive and display a little floating window indicating the number of unread messages.



The Twitteriness of the application is kind of secondary, really. It's certainly nice given the lack of a proper mobile website for the service and the cost (both financial and psychological) of being clubbed by text messages but simply creating yet another delivery mechanism for your friend's idle banter is not going to make anything move beyond your phone.

The thing that I like is that it can double as the rough skeleton to handle the boring bits for any old notification / aggregation agent. Other obvious candidates include recent activity on your Flickr photo stream or recent photos from your contacts using the nifty floating window to display thumbnail versions of the images. But the point is : The phone is connected to the Internets and there is no end to the sorts of things you could do.

(Finally being able to to scratch an
itch <http://www.aaronland.info/weblog/2006/09/28/bake/#pytwitter> *is*
pretty nice too.)

Presumably it won't be long before someone comes along and says this is what [*insert yer syndication format of choice here*] is for. This is true except for the part where it's not. At least not the way feed aggregators are currently designed (and, as I never get tired of pointing out, because Series 60 Python doesn't have a built-in XML parser).

Bouncing "dock" indicators are nothing new on the desktop and it would be easy enough to add similar event handlers to the Series 60 syndication app or even to have it make the phone vibrate. But a calendar is also a kind of aggregator when you get right down to it and few people would suggest, with a straight face, that managing dates and events in a "river of news" is useful or practical. Exaggerating a little, you might say :

"They illustrated the truth that when public housing is at stake, there is not always much to choose from between left and right: Franco's Spain produced the same results as Brezhnev's Russia or Pompidou's France, because abstraction, incuriosity, and greed are among the common attributes of mankind."

— Robert Hughes, *Barcelona* http://isbndb.com/d/book/barcelona_a12.html

(Also : the first one of you to say that a calendar's day view is a "river of news" gets a river of shut the fuck up.)

Your SMS inbox is an aggregator (a literal river of bells and whistles) and mobile carriers have had SMS updates of all flavours for a million years. People have been using both in droves but that doesn't change the part where SMS itself pretty much sucks. People use it in the absence of anything better. Or in the case of many of my own friends and Twitter, not at all.

The de facto syndication framework(s) that have grown up around weblogs are fine but there is more to sharing stuff on the Internet than just shoving everything into the same format. It would be easy to pick on HTML email (or wacky ideas like broadcasting web server logs as RSS) but one has only to look at the never-ending debate about what and *how* to pass around the contents of a feed "item" to understand that one size breaks all.

The adoption of RSS by the weblogging community was a fortunate and convenient hack but it's worth remembering that it wasn't until, what?, last week that Blogger added support for the most basic of syndication elements : titles.

I wouldn't want to leave you with the idea that I think that anything I've done is necessarily *better*. The instant messaging hooks for Twitter have, by all accounts, been a huge success because **they've allowed people to use the service in a way that fits their daily**

habits <http://laughingmeme.org/articles/2006/12/08/twitter-curve> rather than suffering the siren calls of text messaging. Since instant messaging on the

phone is either too painful or too expensive to be useful, I'm just happy that I have the beginnings of the tools to do things *on my own terms*.

See also : **"Like an IM thingy."** <http://www.aaronland.info/weblog/2006/06/12/areacode/#ole> "

This is the part where the conversation rapidly descends into hand-waving and teeth-gnashing rants about doing everything on the web / in HTML (in fairness, Nokia's new browser is a pretty good argument for that; it has become the main reason I don't want to use 2nd Edition Series 60 phones...), the inability of browsers to interact with the rest of the operating system and the best way to scrumjax your phone whether it's using a separate built-in webserver or with some sort of glue between JavaScript and the core APIs. Securely. And abstractly. And easily.

Pick two.

The great thing about being able to see the future on the horizon is that there are so many different ways to try and arrive...

Meanwhile, in the Boring Details department :

The application does not have support for keyboard events (yet) because these are apparently not accounted for in the list of "capabilities" included with the developer's certificate. I'd like to believe that this heralded a near (field) future where I could control my Nokia device-phone with my mind but somehow I doubt that is the case. I *will* get this one sorted out because being able to add a "global" keyboard listener is going to be important to toggle between images, from the default standby screen, for any sort of Flickr application.

Generating a working SIS file is now a three step process : Write the code; "Buildify" the code to squirt in the source for the various shared libraries; Finally generate a signed SIS file.

Or :

```
$> python ~/lib/python/s60-tools/s60-tools.py buildify ./s60_twnotifier.py > ./s60_twnotifier_build
$> ensymble.py py2sis--caps="LocalServices+NetworkServices+ReadUserData+WriteUserData+UserEnvironme
--verbose ./s60_twnotifier_build.py
```

Or :

```
1 | # -*-python*-
2 |
3 | __package__ = "s60_twnotifier.py"
4 | __version__ = "1.0"
5 | __author__ = "Aaron Straup Cope"
6 | __url__ = "http://www.aaronland.info/python/twnotifier/"
7 | __cvsversion__ = "$Revision: 1.6 $"
8 | __date__ = "$Date: 2007/01/21 04:45:49 $"
9 | __copyright__ = "Copyright (c) 2006-2007 Aaron Straup Cope. Perl Artistic License."
10 |
11 | # s60_include : /home/asc/lib/python/s60-simplejson/s60-simplejson.py
12 | # s60_include : /home/asc/lib/python/s60-simplehttp/s60-simplehttp.py
13 | # s60_include : /home/asc/lib/python/s60-simplestore/s60-simplestore.py
14 | # s60_include : /home/asc/lib/python/s60-simplenumbers/s60-simplenumbers.py
15 | # s60_include : /home/asc/lib/python/s60-simpleapp/s60-simpleapp.py
16 |
17 | import time
18 |
19 | class twitter (simpleapp) :
```

Or :

1. **simplejson.py** — *Everytime you use an angle bracket, God kills a member of the W3...I mean, a kitten.* <http://www.aaronland.info/weblog/2006/12/17/meat/#5things>
2. **simplehttp.py** — *Nothing special; just your standard HTTP GET and authentication cruft.*
3. **simplestore.py** — *"We've secretly replaced support for anydbm, in 3rd Edition Python, with support for e32dbm. Let's see if anyone notices!"*
4. **simplenumbers.py** — *Pass it a number and the code will generate a pixelated image of it. This is used in conjunction with the new **topwindow** module to display a persistent hovering window containing the number of unread posts. (It's also a handy indicator of whether or not the OS has killed off the application as Series 60 is want to do without warning...) I wrote it mostly because there's a known bug with 3rd Edition Python builds where text doesn't render properly in*

image objects. And big square numbers. Obviously.

5. *`simpleapp.py` — All the standard set up and take down code for a Series 60 Python app, plus glue for handling notices and error messages.*

6. *`simpletools.py` — Simple. Tools.*

All of which means the "rapid" part of developing applications for Series 60 in Python is kind of a joke but in the grand scheme of things it's probably worth it to keep all that code in separate files.

So, until the final testing and packaging is done :

s60_twnotifier_build.py `/weblog/2007/01/03/itemz/s60_twnotifier_build.py`

2007-01-02T22:45:42-0800



**We'll be right back after a
short Internet meme!**

5 Things You Do For Friends

5 Things You Do For Friends

I hate Interweb memes but I like

Allspaw <http://www.kitchensoap.com/2007/01/18/5-little-known-things/> . You probably already knew that.

- 1. The first thing I ever learned to cook were waffles. I got a waffle maker for Christmas, when I was eight years old.*
- 2. I threw away all my sketchbooks when I was seven years old. I still regret that decision.*
- 3. My earliest memory is of being held in my great aunt's arms eating ice cream, **on the beach*** http://flickr.com/map/?&fLat=41.354685&fLon=-70.649041&z1=4&map_type=hyb . *After that there is an almost two-year blank until my fourth birthday when I got a paper cut from an old-skool computer punch-card. At the time, I could imagine no greater injustice than getting a paper cut on your fourth birthday.*
- 4. In grade two, I turned around for 10 minutes to talk to a friend during which time the only formal schooling in grammar I've ever been a part of was doled out. I'm still never sure who is on first.*
- 5. One day in 1989, I cooked 437 pieces of bacon before 9 AM.*

Now you know.

2007-01-02T22:45:42-0800



bacon:bacon=bacon

Hello, computer.

The Pocketnet

“Wait, aren't machine tags just RDF?”

Hello, computer.



Or : I've posted **version 0.2** of
pmPDF.php <http://aaronland.info/php/pmPDF/pmPDF-0.2.tar.gz> which adds support for
"images <http://flickr.com/photos/straup/389295672/> ".

2007-02-16T08:20:25-0800

The Pocketnet

creases so there's no need to worry about ruler anxiety. You don't even need a printer if all you cared about was having a quick and dirty notebook with pages. Profit!

Their weakness — and this may be by design — is that while their form doesn't limit them to a single sheet of paper a booklet with more than 8 pages requires that the sheets be nested inside one another. Which means you have to keep track of pagination and whether or not page 8 is actually on sheet 1 or sheet 3. The other issue, if you want to generate one of these things using a "computer programming device", is that none of the pages on a given sheet flow left to right; one half flow top to bottom and the other half bottom to top.

Which means that if, like me, you are lazy and have built your tools on **the backs of people smarter than you** <http://www.fpdf.org> you suddenly find yourself up to your arms in someone else's code trying to figure where and when to rotate $\$x$ and remembering that page breaks and automagic new lines are a relative thing depending not only on the actual dimension of your piece of paper but also how many sheets you are using and what "page" you are on.

Which I guess is a really long way of saying thanks to **Aaron** <http://www.flickr.com/photos/x180/276485101/> for goading me into actually doing it :

```
$args = array('folds' => 1);  
$pdf = new pmPDF($args);  
  
$pdf->SetFont('Helvetica');
```

```
$pdf->SetFontSize(18);  
  
foreach (range(1, 16) as $i){  
    $pdf->add_text("page {$i}", $i);  
}  
  
$pdf->Output();
```

Which, let's face it, **is not very exciting** `/weblog/2007/01/24/bacon/pmPDF_example.pdf` . But plumbing never is until you realize that this suddenly doesn't sound quite so crazy, anymore:

By the end of the trip I was full of ideas for community driven websites where you could build and print your day's (or your week's) journey and fold it all in to a handy little booklet that would fit in your back pocket. Like most of these things, it quickly got crushed by too many ambitions and too many moving parts. And then there's the folding thing which is compounded by lack of consumer grade printers to print on anything but letter-sized paper.

It's not quite there yet. A few more boring details remain : Image support and placement (which is actually in the code, half-baked, if you care to look); positioning text in general; more specifically floating text around images. And formatted text which is probably more important to a lot of people than it is to me. But these are details and shouldn't require anything more than the patience to keep track of who is on first. The exciting part is that

this could be the guts (the plumbing) of a system to take the magic of online publishing tools and turn it in to something personal, and maybe even beautiful, that people can hold in their hands unburdened by the weight of technology.

And then **Stikkit went and published their**

API `http://api.stikkit.com/api` .

Maybe I am a bad person for not liking online note taking services. I like Stikkit and continue to be impressed by it, purely as a web application. Despite the fact that I am often tethered to the computer (either by choice or a lack of imagination about what to do with the day) I really don't want to be reliant on the fuckers when I actually need to do or to *remember* to do something. If I scribble something down while I am work, do I really need to whip out my laptop while I am standing in line at the bank machine to see what it was? Besides, if there's no wireless connection nearby and there really is a way to get online **WITH MY MIND** no one has let me in on the secret yet.

4	9	5	7
			Sunday - make coffee - cook duck - hockey game
	todo list #2 - go to 10 - do not pass go	machine tags - make it go - make it go more - go go go - fix typo	Saturday + publish blog entry - add image support - dig out barcode libs
8	1	2	3

So last night, while I waited for dinner to cook, I cranked out a simple library to query Stikkit for my "stuff" and squirt it in to a booklet. You can use the API itself to filter stikkits by tag and date and all the other crunchy goodness they allow and the PDF library to add page breaks after each note. Or not. Suddenly, the Interweb seemed useful again!

But wait — there's more! Imagine that for every note or todo item, there was a **QR (bar)**

code http://en.wikipedia.org/wiki/QR_Code next to it. The code would encode nothing more than a URL containing a link back to Stikkit which would delete the item, or mark it as done. You can do this today (using a **cameraphone with a barcode reader** <http://del.icio.us/tag/barcode+qr>) for stikkits proper but not for todo items since the former action can be done using a plain vanilla HTTP GET request while the latter requires a

PUT.

(This is probably the next thing I will do but it means digging out the barcode library I used for aaPDF, which I promise I will release soon, and writing a bare-bones Atom parser to get the various bits for properly addressing items in the Stikkit database.)

Kellan <http://www.laughingmeme.org/> always scowls at me when I say things like "The whole REST model, purer than the driven snow, is just dumb because of stuff like this" but I am right and he is wrong. There are real security concerns about using GET to modify things on the Interwebs but there are various workarounds (all of dubious quality) and short of encrypted connections and proper crumbs and validation just sending something using the More Better™ HTTP verb strikes me as pedantry. I say this because the street has found its own use for GET; namely, pointers to *do* something. I can encode a proper POST request in a barcode but until I can **invoke a web browser passing that request** <http://www.mozilla.org/docs/command-line-args.html> (instead of a simple URL) we are going to be stuck with problems like this.

In a world where small pieces are loosely joined and **programming anything that fits in your pocket continues to be a chore** <http://www.aaronland.info/weblog/2007/01/03/itemz/#s60widgt> , the built-in web browser will be the only realistic

way to talk HTTP. This is further compounded by the lack of "addressability" in scrumax-y applications. I can't find the post where Stikkit announced that individual notes were made linkable using a GET, but if there's a way to link to an interstitial page to delete an item (for example : `http://stikkit.com/stikkits/nnnnnn/delete`) then I haven't found it.

Assuming I could log in to a mobile-ified version of site proper, passing my API key as a authentication parameter in the absence of a cookie, and do all the necessary POST-y / PUT-y bits in the browser then I would happy enough to do that. So maybe I don't really have any beef with the RESTonauts other than to ask them to remember that, rightly or wrong, it's still a messy world down here.

Details, details. In the meantime, I have released versions 0.1 of **pmPDF.php** `http://aaronland.info/php/pmPDF/` and **pmStikkit.php** `http://aaronland.info/php/pmStikkit/` .

The former accepts one of more chunks of unformatted text (excepting newlines; these things are mostly pointless without the newline love) and generates a PDF file with the text formatted as minty PocketMod pages. The latter uses the former and queries the Stikkit API to generate the text for the final document. Both require the **FPDF** `http://www.fpdf.org` library as a dependency.

The possibilities are endless if by "endless" you mean four or five letter-sized sheets of paper because any more and the whole nesting pages thing starts to get ugly...

2007-02-02T21:10:49-0800

**“Wait, aren't machine tags
just RDF?”**

Meanwhile, in the town called Patience :

No, machine tags are not RDF; they could play RDF
on television, though.

Ladies and gentlemen, **machine tags**

<http://www.flickr.com/groups/api/discuss/721575944978778>:

5/ !

Some day it would be fun to publish the internal "theory
and practice" document we had going while we planned and built
this, but that can probably never happen. As sloppy seconds, I have
started keeping **a list of machine tag related**
pointers <http://del.icio.us/straup/machinetags> on
del.icio.us.

2007-01-24T14:09:04-0800



**“What is the sound of one person
using a platform?”**

TALK TALK TALK TALK TALK

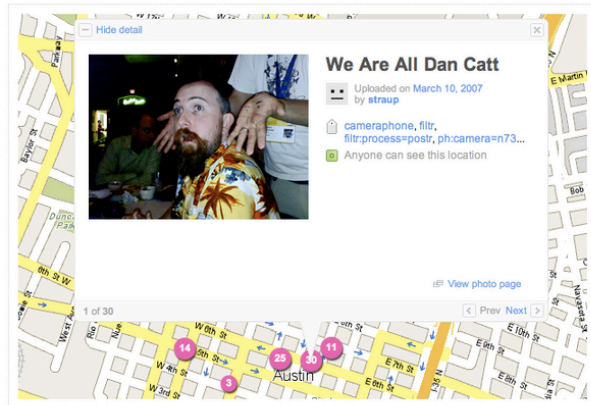
WTF Is On First?

Writing on the wall

I am not a number

TALK TALK TALK TALK TALK

flickr.staff.hire(“geobloggers.com”)



If you're around the Valley tomorrow I will be presenting a talk entitled **"The Scribblenet : APIs, Machine Tags and Magic Words — Building the Do What I Mean Engine"** <http://upcoming.org/event/155488/> , at the Bay Area ACM:

A few quick, sort-of related thoughts :

- *I guess that went well. I have no idea really but I didn't say anything I wouldn't have said over drinks and people kept asking questions until the very end. Neil is probably right* <http://flipzagging.livejournal.com/410859.html> , though.
- *Speaking of Neil, I'm not really sure that I was asking for a "Flickr-style" email application except to say that every single piece of email software out*

there <http://diveintomark.org/archives/2006/06/16/juggling-oranges> would benefit from a "Flickr-style" API. That said, maybe this weekend, I will write something describing how I am archiving and, more importantly, searching email these days. Some time last year I gave up on **trying to use**

MovableType <http://search.cpan.org/dist/MT-Import-Mbox/> for the job and opted for the less painful combination of shell scripts + Perl + Mhonarc + Namazu + a local web server. Apparently this is how the W3C deals with the problem too...

- It stands to reason that the "papernet" is a subset of the "scribblenet". Or something like that. Really the only important thing is : The slide I borrowed from George's talk at SXSW, this year!

Anyway, **here are the**
slides. <http://www.aaronland.info/talks/scribblenet/>

WTF Is On First?

WTF Is On First?

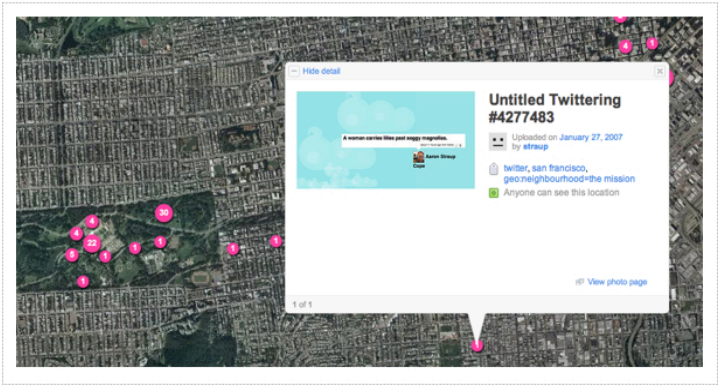
Aaron Straup Cope, SXSW 2007

A Cornell Box With Nothing Inside

“History and geography are what's being thrown away. ... What is geography to Harmony or Pony or Davidson, who speak to people all over the planet every day all at once on their **computer nets** and modems? Or what is history to Mei-Lin or Gaia, who receive seventy-five channels on their families' dish-TV systems? . . . my friends are better prepared mentally for the future that is actually going to arrive.”

Douglas Coupland, *Shampoo Planet* (1992)

I Don't Care Where Starbucks Is

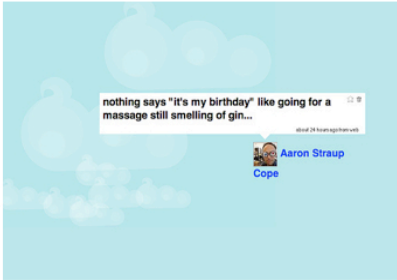


“Units of Ambiance”

“[G]oing on 50 years now we've had the art weenies trying to make a better world by doing performance art in the subway and the urban planners trying to make a better world by herding people in to housing projects and painting the bricks pink.”

Me

“Ambient Intimacy”



Short-term: Cry

**“ALL UR
EXPERIENZ R
BELONGS 2
UZ!!!!”**

Medium-term : *nix Will Rule Mobile

- OLPC
- OS X
- Maemo
- OpenMoko
- Symbia... *whatever*

[this is good]

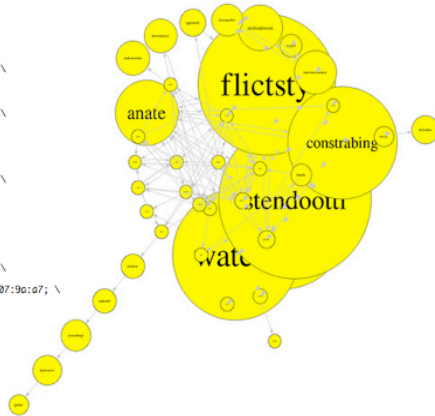
Inter Process Scribble-ability

Medium-term : Learn Python

1. void public class Not(void Java java)
2. GIS
3. DBUS
4. HTTP

“We are all watching each other.”

1154620721;gsm;310;170;240;20291
1154620655;bt;00:0e:ed:19:2a:64;00:13:70:6a:f7:b2; \
00:13:70:6c:27:ce;00:16:db:da:0e:ca
1154620721;bt;00:13:70:6c:27:ce;00:0e:ed:19:2a:64; \
00:13:70:6a:f7:b2;00:16:db:da:0e:ca
1154620787;gsm;310;170;240;20291
1154620787;bt;00:0e:ed:19:2a:64;00:13:70:6a:f7:b2; \
00:13:70:6c:27:ce
1154620855;gsm;310;170;240;20291
1154620921;gsm;310;170;240;20291
1154620855;bt;00:0e:ed:19:2a:64;00:13:70:6a:f7:b2; \
00:13:70:6c:27:ce;00:13:70:6d:24:55;00:07:e0:07:9a:d7; \
00:16:20:84:17:5b



Long-term : Learn JavaScript (Again)

“To verify and demonstrate our concepts we developed a Phonebook mash-up application for S60 devices: The research prototype provides the following functionality: **Access to the native S60 Contacts database**, integration with Yahoo! People Search, and with Google Maps.”

Nokia Research Center

Long-term : Learn JavaScript (Now)

“I created a basic web-based example using the canvas element, and then attempted to ship a simple Cocoa app that wrapped the same WebKit+canvas code. ... This wasn't exactly a revolutionary application, but what was interesting was that all of the diagramming features were implemented using JavaScript, canvas and contentEditable.”

Todd Ditchendorf

DIY

- Track-logs
- Barcodes
- GPS broadcasting
- machine tags and magic words

Dive! Dive! Dive!



Writing on the wall

Part One

My favorite act of abuse is writing in books — and, in this at least, I follow in illustrious footsteps. Mathematics would be considerably poorer were it not for the marginalia of Pierre de Fermat, who in 1637 jotted in his copy of the "Arithmetica" of Diophantus, "I have a truly marvelous proof of this proposition that this margin is too narrow to contain." This casual act of vandalism kept mathematicians out of trouble for 358 years. (Andrew Wiles finally proved Fermat's Last Theorem in 1995.)

Libraries have an ambivalent attitude to marginalia. On the one hand, they quite properly object to people defacing their property. Cambridge University Library as a chamber of horrors displaying "marginalia and other crimes," including damage done by "animals, small children and birds," not to mention the far from innocuous Post-it note. On the other hand, libraries cannot suppress a flush of pride on acquiring an ancient text "annotated" by someone famous. Like graffiti, marginalia acquire respectability through age (and, sometimes, wit).

— **Confessions of a Book**

Abuser <http://www.nytimes.com/2007/03/04/books/review/Schott.t.html> , Ben Schott

Part Two

I never begrudged Simson his not wanting to release the source code to SBook. It was, and remains, an excellent address book but the thing that always spooked me was that you entered stuff as free-form text and application stored stuff as free-form text. SBook would then just *magically* parse the address for you, recognizing names and companies and all (most) of the vagueries of **addressing things** <http://del.icio.us/straup/addressbook> . Which meant that if you ever wanted to do anything with all that data outside of SBook you were left cold with the daunting task of figuring out how Simson built the damn thing.

Then he went and released the source

code. `http://www.simson.net/ref/sbook5/`

I know that Stikkit was not designed to be a general purpose address book.

It has

"peeps" `http://community.valuesofn.com/stikkit/index.php/topic,83.0.html` " but those seem like they are supposed to be more of shortcut, or a view, on the people you may have mentioned in the course of a note. For example, I can add little bits of the same person in different stikkits like this :

Squishy Carrots name:aaron phone:415-555-1212	Shiny Bubbles name:aaron email:bob@example.com
--	---

As different pieces of information are added, the collection is aggregated in a separate stikkit for that person (and, yes, "peep" stikkits really are pink) :

```
aaron  
  
name:aaron  
email:bob@example.com  
phone:415-555-1212
```

This is good. `http://ask.metafilter.com/19916/Meaning-this-is-good` It would be better if when I updated the "peeps" (pink) stikkit the changes

were reflected in all the other stikkits but it is worth remembering that stikkit is not actually an address book.

It's also very easy to confuse the Stikkit address parser (see above) and the vCard export is missing all kinds of stuff (tags and unique identifiers in particular) from the vCard export tool.

Part Three

I mention all of this because I am trying to collect the names and addresses of **bakers in**

Paris [http://www.flickr.com/photos/straup/145251280/in/set-](http://www.flickr.com/photos/straup/145251280/in/set-72057594141964933/)

[72057594141964933/](http://www.flickr.com/photos/straup/145251280/in/set-72057594141964933/) . This raises a few issues, most centered around the part where people (read: me) are lazy :



- 1. It needs to be on the Internets. It needs to have an export API and, ideally, a variety of import endpoints.*
- 2. I'd like to share stuff and let others contribute, as well.*
- 3. But not with everyone.*
- 4. I don't really want to spend a lot of time marking up addresses; ideally I don't want to add any markup at all.*
- 5. It's not really an address book anyway unless you're going to start writing things like "16th St @ corner of don't order the soup".*

This is the part where the Atom people start jumping up and down arguing

that address books are really just "feeds" (a veritable river of who's on first) and everything should be an entry, while the microformat people almost choke imagining everything marked up in hCard.

Just for kicks, **Blaine** <http://www.romeda.org> recommended I store everything in Flickr using **machine** tags <http://www.flickr.com/groups/api/discuss/72157594497877875/> .

Part Four

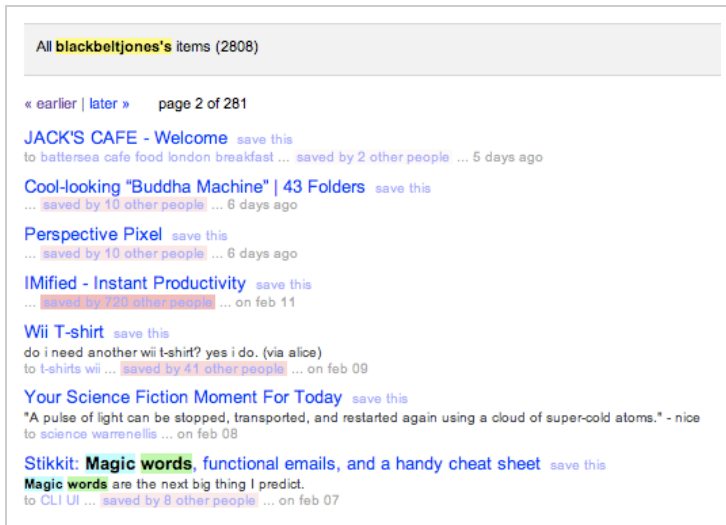
This presentation will review this trend across a variety of textual formats covering the areas above and describe them compared to their XML equivalents with their unique features, goals and limitations and their tradeoffs between simplicity, extensibility and validation. It will also discuss when text formats go bad, causing people to create alternate formats that generate them, when the design fails.

— **<XML/> without the X - the return of {{Textual}}**

markup <http://xtech.expectnation.com/event/1/public/schedule/detail/89> , Dave Beckett

You would be forgiven if you thought this post was about addresses.

Ultimately, it is about "scribblability". It is the part of the "papernet" where people use tools (shared or not) to publish the stuff that interests them. Put another way, it is walking the line between making it easy enough for people to bother putting data in to a system and still useful enough to make it worth the trouble of getting it out.



del.icio.us <http://del.icio.us/tag/aaronland> remains a good illustration of this idea. A lot of time and effort has been spent trying to define a model of describing things, some of it good and most of it either specialized or pendent or both. It's a hard problem but it also tends to be a problem whose solutions, whether it's a methodology or actual living breathing software, are often so complicated and boring as to ensure they are never used. Worse, many are prematurely optimized in to a state of brittleness that prevents any sort of iteration or revision; both reasonable expectations when defining the "**meaning of** **is** http://en.wikipedia.org/wiki/Impeachment_of_Bill_Clinton ".

By way of example, it took a really long time but eventually I realized that the only sane way to label paper file folders was with masking tape.

There's already been plenty of ink spilled on whether or not del.icio.us' tagging system is flexible enough to work with third parties or robust enough to preserve any meaning of value as it grows. For machines. The value of del.icio.us, for me, is the part where it is fantastically easy *for people* to add cues and hints (tags) to, in this case, links leaving the dirty work of assigning meaning for users to sort out *USIN THEIR MINDZ!!!*

Like magic.

Part Five

"Magic" email addresses — addresses that allow you to **send content to a something other than someone's**

inbox http://www.backpackit.com/weblog/archives/tips_tricks/pages_adding_content_to_a_page_via_email.php — are similar in nature and probably

the most useful thing about any web-based service, in a mobile context, while we **wait patiently for better browsers and input**

models <http://www.aaronland.info/weblog/2007/02/17/platform/#barcode>. The messaging application on phones (where there is little if any distinction between email and SMS) are probably the one piece of software designed for speed, at least relative to anything else the device does and the interface is mercifully lacking in bells and whistles. You enter plain unformatted text and not much of it either because you've never figured out how to use "predictive text" of you've bumped up against one of the many limiting agents that dictate the size of a message.

I don't think this is a bad thing, per se. As much as I love tiny computers and their ability to fit in my back pocket they are not always the right tool for the job. I want to write paragraphs on my phone about as much as I want to read anything longer than an essay on any kind of electronic device.

*I once read Bruce Sterling's **The Hacker***

Crackdown <http://www.mit.edu/hacker/hacker.html> "cover to cover" on an old-skool Palm Pilot. It seemed like the right thing to do. It sucked. And hurt my eyes. And my hands. I would love to find a way to make the next 1, 200 page book I read small enough to squeeze in to my jacket pocket — cue the digital ink chorus — but making a book as small as it is large is no solution.

Imagine instead a 2D barcode associated with a subject, let's say a bakery or a restaurant, that encoded a magic email address; assume that the barcode reader is smart enough to recognize the messaging-ness of the address and trigger the correct application for sending-ness. One day, **when barcode readers don't suck**

so hard <http://speedbird.wordpress.com/2007/03/06/off-on-the-wrong-foot/> , you will actually be able to Just Wave™ your phone over a printed barcode and enter **quick notes** <http://www.disambiguity.com/ambient-intimacy/> ("good"; "bad"; "don't order the pork") which are then published back to whatever service the magic email address hangs off of.

Whether or not you go back and edit, or expand on, those notes is moot. The point is you *can*. Or not. Later on, you can print a newer paper version before you head out the door and start all over again.

Part Six

- *The MediaWiki people can't agree on a standard mechanism for editing pages via email* <http://www.gossamer-threads.com/lists/wiki/wikitech/84265> .
- *Stikkit does not expose the magic email address for a page through their API* <http://stikkit.com/api> .
- *The Backpack API does* <http://www.backpackit.com/api/> (but its pages have no "magic" beyond formatting).

Part Seven

Stikkit makes a big deal about "**magic words**" <http://www.43folders.com/2007/02/07/stikkit-introduction/> ". They are to traditional natural language processing (NLP) what "tags" are to capital-O Ontologies. They make things easier.

Back in the days of the command-line interface, users were all Morlocks who had to convert their thoughts into alphanumeric symbols and type them in, a grindingly tedious process that stripped away all ambiguity, laid bare all hidden assumptions, and cruelly punished laziness and imprecision. Then the interface-makers went to work on their GUIs, and introduced a new semiotic layer between people and machines. People who use such systems have abdicated the responsibility, and surrendered the power, of sending bits directly to the chip that's doing the arithmetic, and handed that responsibility and power over to the OS. This is tempting because giving clear instructions, to anyone or anything, is difficult. We cannot do it without thinking, and depending on the complexity of the situation, we may have to think hard about abstract things, and consider any number of ramifications, in order to do a good job of it. For most of us, this is hard work. We want things to be easier.

— In the Beginning was the Command

Line http://project.cyberpunk.ru/lib/in_the_beginning_was_the_command_line/ , Neil Stephenson

Part Eight

There is a limit to computer magic, though, because human language is also magic and computers are still dumb.

If there is anything that the 80/20 Rule doesn't apply to **it is addresses** <http://www.flickr.com/forums/bugs/29641/> . And **recipes** <http://www.aaronland.info/weblog/2006/03/16/terroir/#erdfg> . If a machine makes a mistake with either the results are generally some shade of Bad.

So if we... dumb it down a little and think instead about how to describe — *which really just means writing it down with an expectation of being able to find it later* — **a bottle of**

wine <http://www.aaronland.info/weblog/2006/12/17/meat/#papernet> , we end up with stuff like this:

Traditional (WWW)

Unti	Syrah	2003
------	-------	------

Semi-structured (Wiki)

```
{{Unti|Syrah|2003}}
```

Magic (NLP or "simple")

```
Unti Syrah 2003
```

Magic (Insane)

```
Unti Syrah 2003  
tag as unti, syrah, 2003, wine
```

*For bonus points : **Magic***
(Stupid) `http://www.aaronland.info/weblog/2004/05/18/5490/` .

Part Nine

You tell me which is better.

I've written **the first**
two `http://www.aaronland.info/weblog/2006/09/28/bake/#easy` in the past.
People like Simson `http://www.simson.net/cv/pubs.php` have written the
third. Every one else will probably use the fourth because it's easier (to scribble)
than #2. If, like #3, it's a **losing**
battle `http://radar.oreilly.com/archives/2007/03/another_war_wer.html`
at least it is lost on your own terms.

Now think about

addresses <http://www.aaronland.info/weblog/2006/07/03/smoking/#restobok>, again.

2007-02-20T21:20:02-0800

I am not a number

If you come to see me talk about the

Papernet [http://del.icio.us/search/?](http://del.icio.us/search/?fr=del_icio_us&p=papernet&type=all)

[fr=del_icio_us&p=papernet&type=all](http://del.icio.us&p=papernet&type=all) at **XTech** <http://2007.xtech.org> ,

this year, I promise I will have new and exciting stuff to say. Or, at least, a coherent narrative.

The Good



When I was in

Helsinki <http://www.flickr.com/photos/straup/sets/72157594187100815/> ,

last summer, **Chris** <http://anti-mega.com/antimega/> and I were like a

mutual appreciation club, going on about how useful it would be if people posted

2D barcodes http://en.wikipedia.org/wiki/2D_barcode in public places.

Stores would post their hours of business; public transportation agencies would post schedules at bus stops; restaurants their menus.

Later, **Kellan** <http://www.laughingmeme.org/> and I spent a lot of time on the shuttle bus down to **the mothership** <http://www.flickr.com/map/?>

`&user_id=35034348999@N01&fLat=37.402584&fLon=-121.997251&z1=4&map_type=hyb` bickering about barcodes. Barcodes should encode **nothing more than URLs** `http://semacode.org/` , he'd say.

I think he may have just been trying to get a rise out of me because he usually followed that up with some nonsense about everything being published on the web and microformats and **a world where we do everything online** `http://www.drury.net.nz/2007/02/03/firefox3-web-apps-game-changer/` and the **pipes are never clogged** `http://www.flickr.com/photos/dorkmaster/193674688/` and run pure with magic pixie dust.

It's not total crazy-talk. It allows you to publish (a barcode) once and send stuff on the fly based on circumstances : What time is it? Does the request contain a cookie set during a previous visit? **Is it raining?** `http://search.cpan.org/dist/Acme-Test-Weather/` If a user's device is smart enough to register content handlers for different MIME types, then the URL at the other end of a barcode could send back a calendar or an address book "file" that would be magically squirted in to the right place.

I really only have two problems with that scenario and, in fairness, I am probably wrong about both of them. One of them is the part about "magic" and not squirting but squirting in the "right place". The other thing I don't like is that it means anyone who wants to use 2D barcodes *has* to have a website.

It means that anything I might want to do with a barcode can only happen online which is a little bit like cutting the baby's arms off instead of just swaddling it.

The Non Sequitur



Little pieces of poetry wrapped up in barcodes would be like **easter eggs for the city** <http://www.gpster.net/> .

In practice the whole thing would be horribly abused in about 25 seconds flat but it's also why **you should be able to geotag Twitter posts** <http://flickr.com/photos/straup/370802081/map/?view=everyones> .

I once encoded a recipe for baked beans, marked up as RDF/N3, in a single QR code. It worked, but was several orders of magnitude too dense for anything you might call a "mobile device" to read.

The thing I love the most about **Nokia's barcode reader** http://blogs.s60.com/tommi/2006/05/2d_barcodes_will_rule_the_eart.html is that you can **encode a vCard** <http://www.aaronland.info/weblog/2006/12/17/meat/#paper.net> — their default format for "**business cards**" <http://www.aaronland.info/weblog/2006/08/19/protective/#dwim> " — and it will recognize that the text contains a phone number, a URL, even a postal code but it is not smart enough to figure out that it's looking at a vCard.

There's not really much point in generating QR codes **half an inch or smaller** http://www.qrcodeblog.com/qr/0609/060902_qr_kawasaki_st02.jpg on either side if you're going to try and read them with a cell phone.

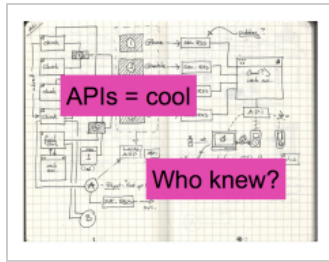
The Bad

There are lots of different kinds of barcodes. So many, in fact, that apparently **"the various stakeholders in the industry must coordinate over some key decisions"** <http://www.mobilecodes.org/> ". Which could be a good thing, given their stated focus on things like "rendering quality", "error correction", whether barcode readers should be stand-alone or built in to a phone's camera application and even boring little details like "aesthetics". By which **they mean to say** <http://www.mobilecodes.org/MC2MS.pdf> :

Outside, in a café, a mobile handset camera is pointed at an advertisement, poster, leaflet or beer-mat. In just one click, the user arrives at a webpage designed specifically for that location. No struggle with the compromised navigational systems of mobile websites; no wait – just the instant fulfillment of the user's needs. The spontaneity of the response encourages an internet connection there and then; the internet content is relevant to the precise time and location of the user; the advertiser can track exactly which piece of paper generated the user response – and the mobile handset has enabled a trouble-free and relevant experience of the web that is potentially more useful to website provider and user alike. And of course, the mobile industry benefits from increased usage of the internet over mobile handsets.

Wow!

It's a good thing I only have to wait until March 2007 to have my every need fulfilled. In a barco...I mean, a beer-mat. My problem with this scenario is not that I, in the course of the last few paragraphs, have suddenly decided that barcodes are a tool of the Man designed to commodif...I mean, fulfill my every need. Rather that the opportunity to do something **useful and exciting** http://www.ironicsans.com/2007/01/idea_paintings_of_descriptions.html is being crushed by the same kind of dumb, lazy and greedy lack of imagination that gave the world WAP.



Why so many people in the mobile industry continue to walk though life thinking that they are the benevolent overlords of happy walled gard...I mean, customer experiences where the *insert your ad here* runs free like beer-mats from a tap is, frankly, a mystery to me. It's a model that, given today's hardware and software limits, has a few years left but is ultimately doomed.

Either way, if people are too stupid to see that or just too greedy to care it is a brutally depressing situation.

Deep breaths.

If you accept the premise that **working code always**

wins <http://flickr.com/photos/tags/qr/clusters/> then the only sane thing to do is to ship your tool, or service, with as many different language bindings as possible or to make it **so brain-dead**

simple <http://tech.groups.yahoo.com/group/rest-discuss/message/7830>

that someone with an itch to scratch will do it for you **over the**

weekend. <http://www.tbray.org/ongoing/When/200x/2003/03/16/XML-Prog>

Which is **why barcodes with URLs will probably make the most**

sense <http://www.aaronland.info/weblog/2007/01/24/bacon/#pocketnet> , at least in the short term. Proper web browsers are either available, in phones, or on their way. It's pretty obvious that the rendering engines will replace the existing GUI toolkits, even if we're a few years off still. When that happens we will have to face the ugly fact that

JavaScript <http://simon.incutio.com/slides/2006/etech/javascript/js->

`reintroduction-notes.html` will finally be the so-called "lingua franca".

from java import acronyms

- . If it's smaller than your laptop
- . And more powerful than a calculator
- . And going to ship with an interpreted language
- . It will probably be Python.

Ultimately, we may still find ourselves in the same boat we are in today `http://www.aaronland.info/weblog/2006/01/09/hirst/#nwtf` .

The problem is not whether the camera application should have an embedded barcode application. The problem is that in the same way people in the mobile industry reacted to the web by creating WAP rather than simply building decent web browsers, they are reacting to the fact that phones have become really small computers by selling "platforms" instead of **fostering the possibilities** `http://opensource.nokia.com/projects/pythonfors60/` .

In fairness, Nokia gets this better than anyone else. Their "**Internet tablets** `http://www.aaronland.info/weblog/2006/01/09/hirst/#edfg2` " are a good example of being able to design and ship a tailored user experience while still allowing developers and hackers and tinkerers the flexibility **to extend it for both fun and profit** `http://maemo.org/platform/docs/tutorials/Maemo_tutorial.html#D-BUS` :

The D-BUS message bus system is used for applications and libraries to deliver messages to each another. ... The message bus is built on the top of a general one-to-one message passing framework, which can be used by any two applications to communicate directly.

But that's just a really long way of saying : They still have shitty cameras and no way to make phone calls.

The Ugly

Or :



Which looks like this:

```
$data = array(
    'street' => '475 Sansome',
    'city' => 'San Francisco',
    'state' => 'CA',
    'zip' => '94111',
    'phone' => '415-555-1212',
);

$addr = "{$data['street']} {$data['city']} {$data['state']} {$data['zip']}";
$url = "http://api.local.yahoo.com/MapsService/V1/geocode";
$url .= "?appid=YahooDemo&location={$addr}";

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$xml = curl_exec($ch);

$doc = new SimpleXMLElement($xml);
$lat = $doc->Result[0]->Latitude;
$lon = $doc->Result[0]->Longitude;

$qqr_geo = new QR(array(
    'data' => '/path/to/qr/data',
    'images' => '/path/to/qr/image',
));

$qqr_geo->encode(array(
    'd' => "http://geo.spum.org/?l={$lat},{$lon}",
    'path' => '/path/to/qr_geo.png',
    'color' => array(69, 139, 0),
));

$vcards = "BEGIN:VCARD\nVERSION:2.1\n";
$vcards .= "TEL;VOICE;WORK:{$data['phone']}\n";
$vcards .= "ADR;WORK;;;{$data['street']};{$data['city']};";
$vcards .= "{$data['state']};{$data['zip']};\n";
$vcards .= "END:VCARD";

$qqr_vcard = new QR(array(
    'data' => '/path/to/qr/data',
    'images' => '/path/to/qr/image',
));

$qqr_vcard->encode(array(
    'd' => $vcards,
    'path' => '/path/to/qr_vcard.png',
    'color' => array(47, 79, 79),
));
```

QR.php <http://aaronland.info/php/qr/> is tiny OOP-ification of **Y. Swetake's excellent qr_img libraries** <http://www.swetake.com/qr/>, with hooks for writing QR codes to disk and specifying a colour other than black. His (her?) release comes with both Perl and Ruby bindings and, at some point, I may sit down and port the code to Python. There is already at least one **QR code generator for JavaScript** <http://la.ma.la/misc/qrcode/>.

The End

You are walking down the street with a big black marker in your pocket

when you see a 2D barcode on the wall. Why *wouldn't* you start filling in the squares?

2007-02-16T19:22:24-0800



Thick like “vegannaise”

pmPDF 0.3

The PaperFS

ICANHASDEVIZEBURGER?

pmPDF 0.3

I've released **pmPDF.php**

0.3 <http://aaronland.info/php/pmPDF/pmPDF-03.tar.gz> with fresh and minty support for variable paper sizes (thanks again to **Theis P.**

Hansen <http://www.valuesofn.com/blog/2007/02/stikkit-in-your-pocket.html#283216311569047619> for the patch) and fixes for the arbitrary positioning of images.

In addition to positioning, the library will also resize images based the available space on the page relative to the image's "starting" point. The tarball contains both example code and a **sample PDF**

file </weblog/2007/04/01/thick/pmpdf03-example.pdf> illustrating the idea.

Here's a screenshot of a more complex version; one which has been touched on in earlier posts :

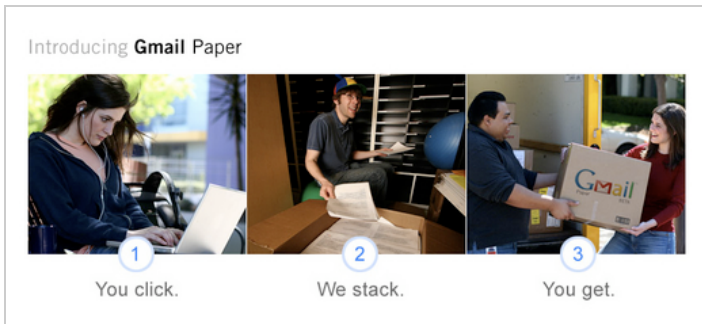
This **one-page booklet** </weblog/2007/04/01/thick/pmpdf03-paris.pdf> was generated from a **text file containing names and addresses** </weblog/2007/04/01/thick/pmpdf03-paris.txt> of bakeries in Paris that was read by a **script** </weblog/2007/04/01/thick/pmpdf03-paris.phps> which plotted the addresses using **Google's geocoder API** http://www.google.com/apis/maps/documentation/#Geocoding_Etc and then fetched various maps using **Yahoo's map image API** <http://developer.yahoo.com/maps/rest/V1/mapImage.html> . For bonus points, it squirted in a variety of **QR codes** <http://www.aaronland.info/weblog/2007/02/17/platform/#barcode> and then laid it all out in handy **PocketMod** <http://www.aaronland.info/weblog/2007/01/24/bacon/#pocketnet> format.

By which I mean to say : When do we get APIs for **My Maps** <http://www.aaronland.info/weblog/2007/04/01/thick/#paperfs> ?

The PaperFS

The Joker

Like everyone else, I laughed when Google "announced" their **Gmail Paper** <http://mail.google.com/mail/help/paper/more.html> service, on April 1.



The thing about the joke, though, is that it's a *really good* idea.

Sooner or later everyone starts to think about generating a paper dump of their 4-billion email messages or their weblog posts. No one does it, but everyone thinks about it.

The Armchair Quarterback

I had an **80/20 slide** <http://aaronland.info/talks/sxsw07/#slide2> in the presentation I did at SXSW, this year. 20% of the slide was just an excuse to wrap the phrase "computer nets" in a pink <blink> tag. The other 80% was to point out an idea that has remained popular since it was first written around the time the Internets were being loosed from Universityland :

History and geography are what's being thrown away. ... What is geography to Harmony or Pony or Davidson, who speak to people all over the planet every day all at once on their computer nets and modems? Or what is history to Mei-Lin or Gaia, who receive seventy-five channels on their families' dish-TV systems? . . . my friends are better prepared mentally for the future that is actually going to arrive.

Douglas Coupland http://isbndb.com/d/book/shampoo_planet.html

The quote was a setup and the punch-line was : This is wrong.

Geography has never not been important and I think this is one of the reasons why people have responded so positively to

Dopplr <http://del.icio.us/mattb/dopplr> . Going some *place* still means

something because there is more to human relations than action items and emoticons. We still live in a world where we are subject to physical constraints — mostly the constant tension involving **the**

weather <http://vimeo.com/clip:152534> and individual temperaments — and we can't all be in the same place at the same time.

Also : Drinking with people in virtual environments is a pale shadow of the real thing. <http://vimeo.com/clip:152223>

To succeed on Hudson Bay, the Inuit needed to know everything about their immediate surroundings: the landmarks, the animals' travel and migration routes, the location of fresh-water springs, berries, bird eggs and willow-worm cocoons to dip into seal fat for dinner. Describing the land's natural features with lyrical precision, McGrath emphasizes that the harsh physical realities of this place shaped not only how the Inuit lived but also their personalities, making a strong case that psychology is destiny. At one time, expressing rage, lust or ambition were considered so threatening to Inuit group survival that persistent offenders were banished.

Elizabeth Royte <http://www.nytimes.com/2007/04/08/books/review/Royte.t.html>

Geography is how we "frame" the moment and we should enjoy the still nascent ability to both easily communicate *and* travel long distances while it is **still morally and economically sustainable** <http://www.hackdiary.com/archives/000096.html> .

History is what gives a moment its nuance and it is tempting to imagine it as, or becoming, a black and sexy William Gibson-esque carbonite **bag-of-holding** http://en.wikipedia.org/wiki/Bag_of_holding but we all know that's a crock of shit. Paper is not without its flaws but the evidence is pretty overwhelming that it remains the best 80/20 archiving system we've developed to date.

And the first person to jump up and down yelling "Fire at Alexandria! Fire at Alexandria!" can shove it up their 5 1/2 inch floppy drive. Or their Zip drive, SCSI or parralel; take your pick. Or that DVD you bought yesterday for 0.15 \$ which is probably already corrupted.

There's a reason that Google is busying themselves with **optical character recognition (OCR)** <http://del.icio.us/tag/google+ocr> software. After all, written languages are just "barcodes" that humans happen to understand.

The Storyteller

Enter **user-defined**

places <http://flipzagging.livejournal.com/412146.html> , aka the fantastically badly named "MyMaps". Complete with a **print**

button <http://ylocalblog.com/blog/2007/04/06/new-interactive-print-page-for-maps/> .

2007-04-06T09:02:46-0700

ICANHASDEVIZEBURGER?

The very kind **Theis**

P. <http://www.valuesofn.com/blog/2007/02/stikkit-in-your-pocket.html#283216311569047619> recently sent me a patch to allow for variable page sizes in my

Papernet/Pocketmod <http://www.aaronland.info/weblog/2007/01/24/bacon/#pocketnet> libraries. **If I were a better person I would applied it by**

now. <http://www.aaronland.info/weblog/2007/04/01/thick/#pmpdf03>

Instead, I got Series 60 Python hacking on the brain last week so on the mornings when I could do more than stare vacantly at my coffee I cobbled together a couple quick and dirty applications.

GPSRecorder

Or : If you think that you've already seen this done before in **nmea_info.py** <http://gagravarr.livejournal.com/104793.html> , you'd be right.

Nick has written an excellent piece of code complete with the hooks to write log files, take pictures and **squirt GPS data directly in the EXIF headers** <http://benno.id.au/code/pexif/> . Unfortunately it didn't Just Work when I ran it on my 3rd Edition phone and I was too lazy to start digging through someone else's code to find out why.

The other problem with trying to write any kind of camera application on 3rd edition phones is that the devices all have lense coverings whose hardware is set to launch the camera application as soon as they are opened. I gather that you can get around this, or define your own triggers, if you are writing programs in C++ or Java but no such love exists in Python land.

And all I wanted was something that would sit in the background talking to a Bluetooth GPS device recording a latitude, a longitude and a timestamp. So that's *all* it does, writing tab-delimited values to text files (stuff is grouped by year-month-day) for transfer and processing at the time of your choosing.

Profit!

I also wanted something that was light and simple which could be easily used in other applications; by which I mean to say I need to further pull things apart to have a generic `s60-simplegpsrecorder` library, but one thing at a time...

For example, the nice people at [geonames.org](http://www.geonames.org) have a **whole set of reverse-geocoding API methods that are keyed off latitude and longitude** <http://www.geonames.org/export/reverse-geocoding.html> . This is fantastic because it means you can establish an actual context for a location beyond "data-points" which, while useful, aren't the kind of stories that speak to the heart.

Mercifully, in a world where **Nokia still hates XML** <http://www.aaronland.info/weblog/2006/09/28/bake/#pytwitter> the geonames people also elected to make JSON a response format so you can use it with Series60 Python.

(**Dan Catt's discussion of mashing together geonames, triplr and JSON** <http://geobloggers.com/archives/2007/03/31/geonames-rdf-triplr-json-yahoo-pipes-and-the-semantic-web-oh-my/> is very much related, in a twisty maze kind of way, and definitely worth a read.)

Which is all good since it means that maybe I can change the elevator version of **the talk I gave as part of Dan's mapping panel at SXSW** <http://aaronland.info/talks/sxsw/> to be just "doom... doom... doom..." instead of "DOOM!! DOOM!! DOOM!!!"

Speaking of JSON-based APIs, that brings us to :

Twitteradio

No, really.



This one just sits around asking Twitter for new posts via the **public timeline API** http://twitter.com/statuses/public_timeline.json and then reads aloud them using Python's

audio <http://discussion.forum.nokia.com/forum/showthread.php?t=95141> hooks. You know, sort of like :

```
perl -e 'while (1) { system("/usr/bin/say 9"); }'
```

It is just as annoying as it sounds. Actually, you can't really imagine *how* annoying it sounds until you've heard the default Text-To-Speech voice that ships with 3rd Edition phones. But it remains :

AWESOME! <http://flickr.com/photos/hitherto/298258283/>

Especially when it says "**Matt Jones** <http://www.vimeo.com/clip:162986>".

I expect to see twitterradio in a **dance**

remix http://www.waxy.org/archive/2002/07/21/the_emin.shtml , near you, soon enough.

"There is a special place in Hell for the likes of you."

I've bundled up each application as a semi-standalone SIS file for "easy" installation but they still require that you have a recent copy of **Series 60**

Python <http://opensource.nokia.com/projects/pythonfors60/> installed on your phone. I will get around to building proper standalone applications shortly.

As with most Series 60 Python apps, whether or not the SIS file will work on *your* phone is a bit of a crap shoot, especially if you are using a newer 3rd edition "device". I've included the source for the bored, courageous and frustrated. All I can tell you is : They work for me.

GPSRecorder should work on both 2nd and 3rd Edition phones.

- ***GPSRecorder 1.0 SIS***

file http://aaronland.info/bin/s60_gps_recorder/GPSRecorder-1.0.sis

- ***GPSRecorder source***

code http://aaronland.info/bin/s60_gps_recorder/s60-gps_recorder_build.py

Twitterradio requires a 3rd Edition phone running Series 60 Python 1.3.14, or higher.

- ***Twitterradio 1.0 SIS***

file http://aaronland.info/bin/s60_twitterradio/Twitterradio-1.0.sis

- ***Twitterradio source***

code http://aaronland.info/bin/s60_twitterradio/s60_twittera

`dio_build.py`

IM IN UR PANTZ CONVERGIN UR SYNERGEEZ!!!

Automagic

twitvision `http://radar.oreilly.com/archives/2007/03/twitvision_a.html` posting. I'm just saying.

Update:

By which I mean to say : Here's a rough around the edges application that does almost everything described above. It doesn't record anything for later processing but hangs around and asks the BT/GPS device for a latitude and longitude and then asks **geonames** `http://www.geonames.org/` where that point actually *is*.

Then the information is printed on the screen which is little like checking the Internets to find out the weather outside but all I can is that it was a proof of concept. A more interesting thing to do, maybe, would be to pass the latitude and longitude to the **Yahoo! Map Image API** `http://developer.yahoo.com/maps/rest/V1/mapImage.html` and display a pretty map. This is on the list.

Instead, I added a "Tell Twitter" menu item. As in :

I've also updated the code-y bits such that if you are trying to run this on the desktop with the **PyS60 emulation libraries** `http://sourceforge.net/projects/pys60-compat/` it will do the BT/GPS dance assuming you've also installed the excellent **Lightblue Bluetooth libraries** `http://developer.yahoo.com/maps/rest/V1/mapImage.html` .

These should be both be considered betaware and all the usual caveats apply. Once they've been given a little more polish I will give them a permanent

home...

- ***GPS Locator SIS***

file `/weblog/2007/04/01/thick/s60_gps_locator_build.sis`

- ***GPS Locator source***

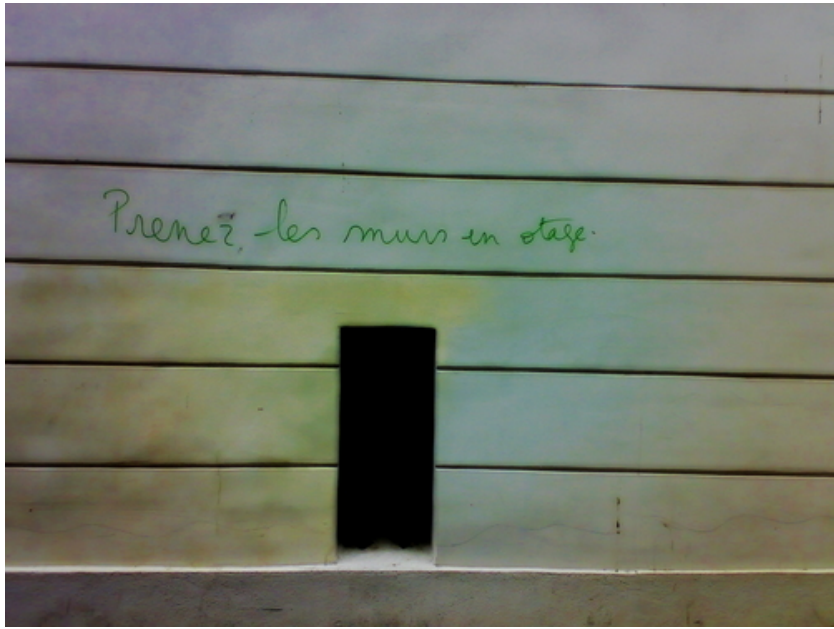
code `/weblog/2007/04/01/thick/s60_gps_locator_build.py`

Because information wants to tell you what it's doing. By which I mean to say : Once I've gotten

Log::Dispatch::Jabber `http://search.cpan.org/dist/Log-Dispatch-Jabber/` working again, I will probably be forced to write

Log::Dispatch::Twitter...

2007-03-30T07:52:54-0700



Donde esta la playa???

Once I was lost but now I am still not found

mmPDF.php 0.0 0.1

re-hello world

The Papernet

**Once I was lost but now I am
still not found**



Tim

Bray <http://www.tbray.org/ongoing/When/200x/2007/05/31/Gone-World> pointed out a mostly incomprehensible wandering thought about Toronto, **by William**

Gibson <http://www.theglobeandmail.com/servlet/story/RTGAM.20070530.wlum-gibson0531/CommentStory/luminato/>, the other day. As best I can make out "Toronto" is just a device to make the larger point that the 20th century was unkind to urban planning — *cue the chorus of ninnies whining "Jane Jacobs! Jane Jacobs! Jane Jacobs!"* — and the 21st century doesn't look much better.

And something about how Toronto was better before anyone knew it existed and rent was still cheap. One can only

assume that the phrase "*Montreal I at least had heard of.*" is really code for : I blame Toronto's freakish obsession with being a "world class city" on the election of the Parti Québécois in 1976.

Anyway, the one part that stood out for me was the idea of "gone worlds" :

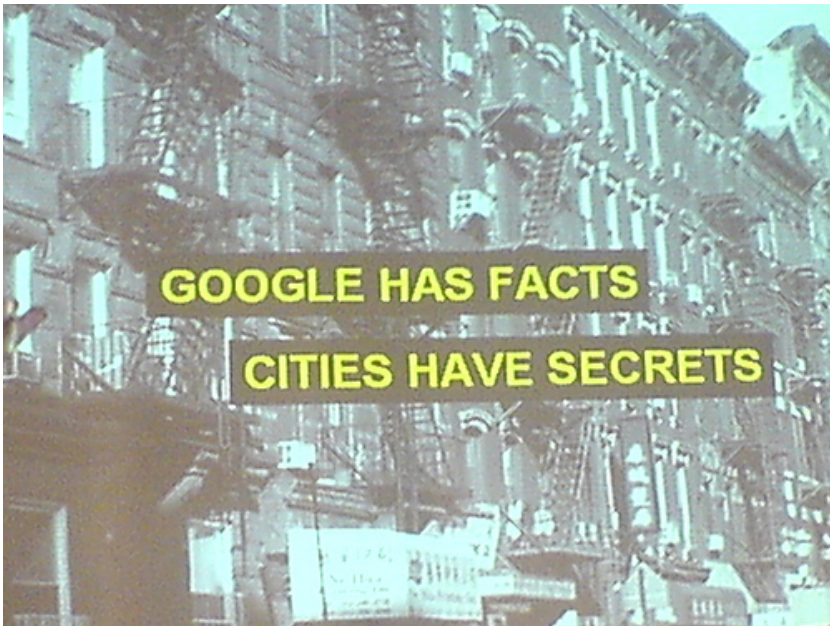
A friend of mine in New York has been pointing out surviving bits of what he calls his city's "gone world" to me for the past 20 years or so. When I first started getting to know New York, in the early 1980s, it consisted mostly of that gone world, or so it seemed to me. People who lived there didn't seem to believe it possible that this would change. My friend was the first New Yorker I knew who noticed that things there were changing, becoming gone.

The sewing machine spare-parts quarter, for instance (gone), or the tenement that once housed McGurk's Suicide Hall (gone). Bits and pieces of SoHo and TriBeCa and Chelsea, all gone. Had I not had so observant a guide, I certainly would have missed them, these glimpses of vanishing things, but my friend had treasured them all, and was pained by their going, and took care to show them to me. It was his conviction that they were invariably replaced by much less interesting things (to put it mildly), and I generally agreed.

If I ever do **the Papernet**

talk <http://aaronland.info/talks/papernet/> again I am going to mention "gone worlds" because they are an interesting point from which to think about things like travel journals and personal publishing and, out at the more dysfunctional end of the scale, "**Bag of**

Holding http://radar.oreilly.com/archives/2007/05/where_20_iphone.html " syndrome.



Probably somewhere after the "**Douglas Coupland was wrong** <http://flickr.com/photos/pip/507904022/in/set-72157600239267901/> " slide since that's a horse never too old to flog.

2007-06-02T05:44:20-0700

mmPDF.php ~~0.0~~ 0.1

Date: Mon, 28 May 2007 16:21:54 -0700
From: **Aaron Straup Cope**
To: **Chris Heathcote**
Subject: Proof of concept : mmPDF

I seem to have run afoul of Y! Map's DOS system so there aren't any actual *maps*.

But it's still pretty cool considering that the **attached attachment** was generated with nothing but the following :

```
$pdf = new mmPDF($args);  
$pdf->SetFont('Helvetica');  
$pdf->SetFontSize(10);  
$pdf->add_map("IM IN UR MAPS STEALIN UR PARIZ!!!!");  
$pdf->Output();
```

I will send you a proper sample just as soon as I get to another IP address...

Update: The good news is I got it to work. The bad news is that Chris' comprehensive list of Paris eateries results in a 16MB PDF file. Whether or not there are optimizations to be made (converting all those PNG files into JPEGs maybe...) remains to be seen.

The open question is : A 208-page (that's 26 sheets of US letter-sized paper) PocketMod book? But you get the idea. I've uploaded a shorter, smaller **40-page (5 sheets) sample document** </weblog/2007/05/21/playa/mmPDF-0.1.pdf> that includes the first 20 entries in **Chris' list** <http://maps.google.com/maps/ms?ie=UTF8&hl=en&msa=0&msid=100299907049452789878.000001128c15a3e9d7622&z=13&om=1> . Since this is a 0.1 release, here's a short list of things this package *doesn't* do yet :

- *Indexes or table of contents.*
- *Maps scaled (read: tested) for anything but 8.5 x 11*

inch sheets of paper.

- *The ability to place more than one location on a "page"; this could be useful if you were printing on, say, a broadsheet.*
- *The ability to specify only a single map per location (entry).*
- *Magic to take a standard KML "network link" and just sort out where the actual data is.*

Say hello to **mmPDF.php**

0.1 <http://aaronland.info/php/mmPDF/mmPDF-0.1.tar.gz>

2007-05-28T16:26:17-0700

re-hello world

I'm back

Let's just leave it at that. When I first wrote this post, yesterday morning, I had successfully managed to forget that people walk around in public wearing those stupid Star Trek phone-things in their ears. Then I took the BART downtown.

Slides from my talk on the “The Papernet”

I've put **the slides from my talk at XTech**

online <http://www.aaronland.info/weblog/2007/05/21/playa/#papernet> . There are many, many images most of which have not been squished for the web so be patient. It will look fugly for a bit and then the magic JavaScript will kick in render stuff all nice and pretty.



I don't really feel like diving back in to the code for

S5 <http://meyerweb.com/eric/tools/s5/> but lazy loading of images would be a nice thing and, as Neb points out, it's not very

good about wrapping text or resizing images on the fly. I understood that the projector would automatically switch in to 1024x768 mode but when I got up to the podium my world quickly shrank to 800x600 pixels. Maybe it's just a switch (in S5) somewhere...

There's also **a video of the talk available online.** <http://cubicgarden.blip.tv/file/229942/> I haven't watched it yet but, apparently, that's what I said. Towards the three-quarter mark, there's a mention of a friend using MyMaps to do almost everything I've been talking to up until that point. **That would be Heathcote.** <http://antimega.textdriven.com/antimega/2007/05/15/holidaying-fooding-talking>

Meanwhile, I am scheduled to be in London in June for **Hack Day UK** <http://www.hackday.org/> to share with you the magic of **machine tag love** <http://yahoresearchberkeley.com/blog/2007/05/16/the-emerging-semantics-web-the-semantic-web-is-dead/> .

There's more still to write about XTech, which was excellent <http://flickr.com/photos/bopuc/505076855/> , *but I will save that for later.*

s60-simplelibs

You have **Matt**

Webb <http://interconnected.org/home/> to thank for motivating me to finally finish bundling up all the various helper libraries I've been using when writing S60 Python applications. From the README :

These are a series of simple wrapper libraries designed to make developing Python applications using Series60 3rd Edition suck less.

The simplelibs package is just a bundled distribution of other packages and should not be invoked directly in your code.

As of this writing it does not have a proper setup.py file and there is almost no documentation. There are probably also bugs.

Ladies and gentlemen, **s60-simplelibs**

0.1 <http://aaronland.info/python/s60-simplelibs/s60-simplelibs-0.1.tar.gz>

filtr 0.23

These are mostly tweaks to make generating photos using the "**heathr**" <http://flickr.com/search/?w=35034348999%40N01&q=heathr+barcelona+&m=tags> " filtr better.

Specifically, surrounding borders on the final canvas are no longer used and, because the input files are modified, temporary copies are created and used in their place.

And then I said : **filtr**

0.23 <http://aaronland.info/bin/filtr/filtr-0.23.tar.gz>

Paris

horse=yes! <http://flickr.com/photos/straup/sets/72157600224645171/>

Barcelona

I love
you. <http://flickr.com/photos/straup/sets/72157600202022198/>

2007-05-21T07:52:45-0700

The Papernet

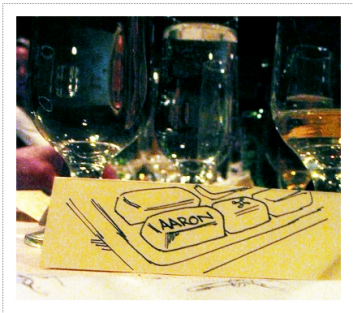
*In 2007 I did not prepare notes for talks which was always a bit risky. I should have but given how much had, and continues to be, written about **the***

papernet <http://www.aaronland.info/papernet/> *on this weblog maybe it's moot? Anyway, here are the slides and **there is also video of the talk*** <http://blip.tv/cubicgardencom-videos/semantic-meaning-of-everyday-objects-in-a-connected-world-234364> .

The Papernet

Aaron Straup Cope
XTech 2007, May 15 2007

Hi, My Name Is Aaron



Recipes



Index Cards



Ubiquitous Does Not Mean “Always On”

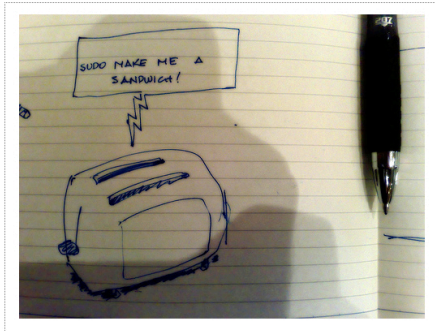
eatdrinkfeelgood.org
Below are sponsored listings for goods and services related to: eatdrinkfeelgood.org

Search

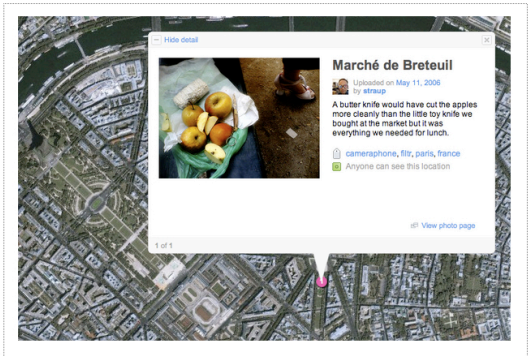
Sponsored Listings
[High Speed Internet](#)
Available in Burton CenturyTel Think Fast- \$19.95/month
[www.CenturyTel.com](#)
[Land near Burton, Texas](#)
Pre-Development Pricing, 1-3 Acres. Premium Wooded & Hilltop Community
[www.KingOaks.com](#)

Related Searches
[Feelgood](#)
[Eat Drink](#)
[Motley Crue](#)
[Feelgood](#)

The Revolution Will Not Be Convergent



Paris Stories



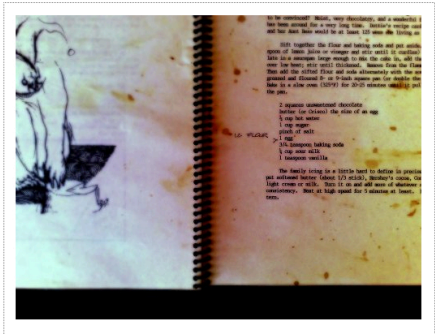
“Micro-Publishing” (Like, Oh My God!)



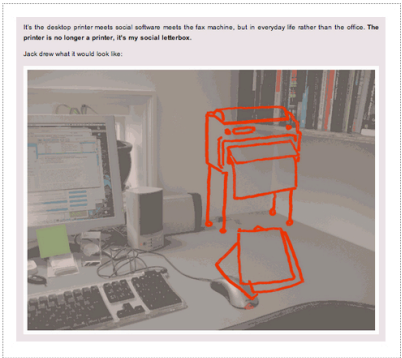
bagOf(Holding(Memory(Sticks)))



Artifacts Are the Soft-Porn of Memory



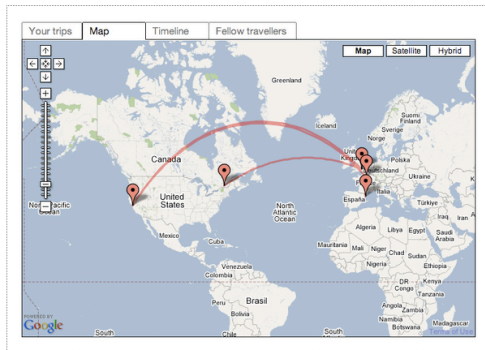
No. Really.



Douglas Coupland Was Wrong



No. Really.



“Hello, Computer!”



E_EXCESSIVE_IO

*“[W]alking the line between making it **easy enough for people to bother putting data in** to a system and still useful enough to make it **worth the trouble of getting it out.**”*

Translation

*“There is a limit to computer magic because **human language is also magic** and computers are still dumb.”*

Translation



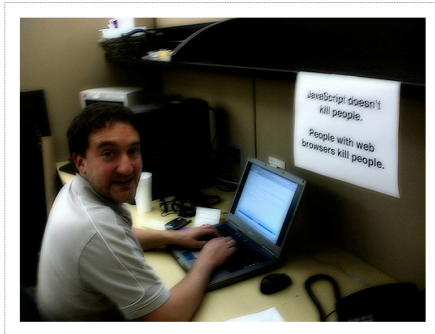
Translation

DOOM!!!

DOOM!!!

DOOM!!!

Desktop (The Web Is Not Your)



Mobile



Barcodes



Magic Words



Magic : Traditional

Fleur Petite Syrah 2003

Magic : 2.0

```
fleur:petitesyrah=2003
```

```
vin:producer=fleur  
vin:wine="petite syrah"  
vin:vintage=2003
```


Magic : Classic


{{wine|Fleur|Petite Syrah|2003}}

Magic : Insane

Fleur Petite Syrah 2003

tag as fleur, syrah, 2003, wine

Magic : Must Die



Recipes - edit item

Title

Details (optional)

Details are attributes that define the item and help searchers find it. For example, "Author: Ernest Hemingway" or "Area: 450 square km".

Main ingredient:

Cuisine:

Cooking Time:

Attribute name

Attribute value

Text

Author: Joe Smith

Number/unit

Memory: 2 gigabytes

Number

Bedrooms: 2

Date range

Birthday: 6/14/1965

Large text

Directions: Turn right at Main Street, ...

Web URL

Home page: http://www.google.com

Checkbox

Parking space included?

Location

1 Main Street, Anytown, USA

Maximum of 10. Separate with commas.

Translation

DOOM!!!

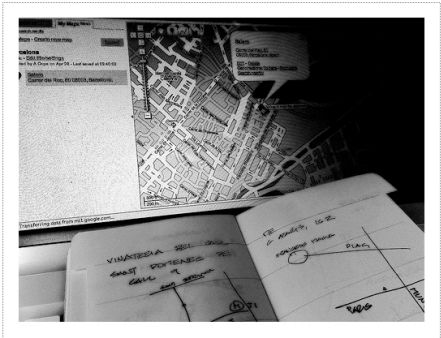
DOOM!!!

DOOM!!!

Or Not.



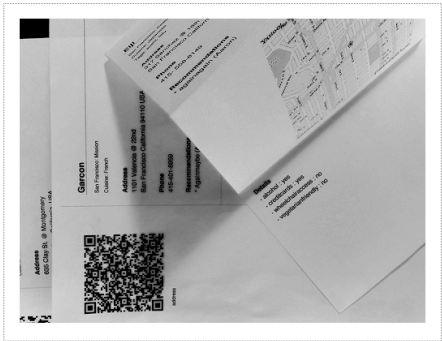
No. Really.



Is That A Computer In Your Pocket?



Origami For The Paperless Office



grape.spum.org

```
{{wine|Unti|Petit Frere|2004|red}}  
{{grapes|Grenache|Mourvedre|Syrah}}  
{{region|USA|California|Dry Creek Valley}}  
  
= Recommendations =  
  
{{recommendation|Aaron|again}}  
{{recommendation|Mishu|good}}  
  
= Points of Sale =  
  
{{pos|Bi-Rite}}
```

grape.spum.org

Name

[Unl'd Petit Fraise 2004](#)

Grapes

- [Grenache](#)
- [Mourvedre](#)
- [Syrah](#)

Region

- [Dry Creek Valley](#)
- [California](#)
- [USA](#)

Recommendations

- [Aaron](#) says "again"
- [Mishu](#) says "good"

Points of Sale

- purchased at "[B-Site](#)"

grape.spum.org

Facts about Unti Petit Frere 2004 — Click +  to find similar pages.

Relations to other articles

Produced by	Unti ± 
Has label	Petit Frere ± 
Bottled in	2004 ± 
Is a	wine ± 
Has colour	red ± 
Has grape	Grenache ±  , Mourvedre ±  , and Syrah ± 
From locality	Dry Creek Valley ± 
From region	California ± 
From country	USA ± 
Aaron's recommendation	again ± 
Has recommendation	again ±  , and good ± 
Recommendation from	Aaron ±  , and Mishu ± 
Mishu's recommendation	good ± 
Has point of sale	Bi-Rite ± 

Category: [Wine](#)

grape.spum.org

Wines from California

	produced by	from locality	has recommendation
Moshin Vineyards Pinot Noir Clone 115 2004	Moshin Vineyards	Russian River Valley	Again
Fleur Petite Syrah 2003	Fleur	Central Valley	Again
Pavi Dolcetto 2004	Pavi	Napa Valley	Again
Noceto Sangiovese 2004	Noceto	Plymouth	Againmaybe
Unti Petit Fren 2004	Unti	Dry Creek Valley	Again Good
Unti Barbera 2005	Unti	Dry Creek Valley	Againmaybe
Unti Sangiovese 2004	Unti	Dry Creek Valley	Againmaybe
Fritz Zinfandel 2004	Fritz	Dry Creek Valley	Bad
Rabbit Ridge Syrah 2005	Rabbit Ridge	Dry Creek Valley	Bad Unexceptional
Fritz Pinot Noir 2004	Fritz	Russian River Valley	Boring
Russian Hill Ellen's Block Syrah 2001	Russian Hill	Russian River Valley	Great

grape.spum.org

Look for wines

where

region

is

california

and

recommendation

is

again

and

search

Options

Close

Unti Petit Frere 2004

Aaron says "Again"

Mishu says "Good"

Moshin Vineyards Pinot Noir

Clone 115 2004

Aaron says "Again"

Pavi Dolcetto 2004

Aaron says "Again"

Unti Sangiovese 2004

Aaron says "Againmaybe"

Options

Back

grape.spum.org



DCK (1912 - 1987)



2007-05-18T07:52:45-0700



Mining for Pynchonite

Writing filters for Namazu, in a nutshell

```
Net::Flickr::Geo->tease()
```

FIREBAGEL!!!!!!

```
Net::Flickr::Geo
```

Writing filters for Namazu, in a nutshell

Some time last year I gave up trying to **use Movable Type to archive my email** <http://aaronland.info/perl/mt/import/mbox/> . Instead, I've been using **MHonArc** <http://www.mhonarc.org/> and **Namazu** <http://namazu.org/> , a full-text indexer with both command-line and web interfaces, plus some fancy-pants templates and bookmarklets to make searching easier and prettier.

As usual, at some point I will get around to making all of that stuff public. But not today. If for no other reason than that it is written to expect that you store your email in nested (YYYY/MM/DD) directories which apparently the rest of the world thinks is "weird".

Which is context for saying that you can use Namazu to index just about anything, especially if you can write a custom **"filter"** <http://www.namazu.org/doc/manual.html#doc-filter> " to extract, infer or generate data from an arbitrary file. Somewhere on my TODO list is a proper search interface for this weblog.

Except for the part where what little **documentation there is for writing filters** <http://www.namazu.org/~kenji/dekiru-namazu-filter.html.en> generally leaves you asking more questions than when you started.

So, in an effort to give something back to the Interweb here is the quick and dirty guide to writing Namazu filters. This is by no means a complete reference and it may not contain all the

information you need but it does contain everything I needed this morning (and am likely to forget by the end of the week).

1. *Filters are written in Perl. Deal with it, or at least learn enough to use the **open** <http://perldoc.perl.org/functions/open.html> function.*
2. *The stuff that finally gets indexed by Namazu is stored in the **\$cont_ref** variable which is, well, a scalar reference.*
3. *You can override the list of default "fields" that a user can query on by (re) setting **\$conf::META_TAGS** (and using the **-M** flag when building your index) and storing their values in **\$fields** which is a hash reference.*
4. *Do yourself a favour and define an "x-type" for whatever you're trying to index and force it, using the **-t** option, when you are building your index. This is especially useful when you are crunching XML files since they seem to be handled by the **HTML** widget by default.*
5. *If you return anything other than **undef** from the "filter" function, Namazu will assume it is an error string, spew it to **STDERR** and not index the file.*

That's pretty much it. An actual filter package looks something like this :

```
package bucketz;
require 'util.pl';

# A bunch of hooks that at
# least need to be present...

sub status { "yes"; }
sub recursive { 0; }
sub pre_codeconv { 0; }
sub post_codeconv { 0; }
sub add_magic { undef; }

sub mediatype {
    return ("application/xml; x-type=BUCKET");
}

sub filter {
    my $orig_cfile = shift;
    my $cont_ref = shift;
    my $weight_str = shift;
    my $headings = shift;
    my $fields = shift;

    my $cfile = defined $orig_cfile ? $$orig_cfile : '';

    if (! &icanhas($cfile)){
        return "FAIL!!!!!!";
    }

    my $extra = join("|", ("bucket", "cheezburger", "meme"));

    if ($conf::META_TAGS !~ /$extra/){
        $conf::META_TAGS .= $extra;
    }

    # DO STUFF WITH $cfile, $cont_ref and $fields here...
    # DO STUFF WITH the other variables; that's your business

    $$cont_ref = "OH HAI";
}

return 1;
```

Once you've installed the filter in a place where Namazu can find it you need to rebuild your index like so :

```
$> mknmz -M -a -t 'application/xml; x-type=BUCKET' -O /yer/index /yer/filez
```

By default, Namazu is set up to index plain text and only allows you to define email-style fields; From, Subject, etc. The

plain text part isn't going to change but the ability to write your own converter and to also add custom limiting agents (fields) allows you to *[insert obligatory Flickr tags hack here]*.

Or :

```
$>namazu '+place:russia' ~/news/nytimes/.namazu/  
Results:
```

```
References: [ +place:russia: 2 ]
```

Total 2 documents matching your query.

1. The Kremlin Flexes, and a Tycoon Reels (score: 1)

Author: Andrew Kramer

Date: Sun, 08 Jul 2007 10:44:11 -0800

The end of a partnership in the world's largest nickel producer illuminates how the Kremlin and ambitious Russian businessmen do business together.

<http://www.nytimes.com/2007/07/08/business/yourmoney/08nickel.html> (21,957 bytes)

2. Youth Groups Created by Kremlin Serve Putin's Cause (score: 1)

Author: Steven Lee myers

Date: Sun, 08 Jul 2007 10:44:10 -0800

A youth movement seeks the ideological cultivation, some say indoctrination, of the first generation to come of age in post-Soviet Russia.

<http://www.nytimes.com/2007/07/08/world/europe/08moscow.html> (12,788 bytes)

Now you know.

2007-07-08T13:12:58-0700

Net::Flickr::Geo->tease()

You know the thing with the maps and the tiles? And the other thing with the math? And then the part where you have to keep track of who's on first because no one does it the same way and every provider does one map version better than the others?

Well, **Mike Migurski** <http://mike.teczno.com/> loves you :

```
437 |sub fetch_modest_map_image {
438 |    my $self = shift;
439 |    my $lat = shift;
440 |    my $lon = shift;
441 |    my $acc = shift;
442 |
443 |    my $path_composer = $self->{'cfg'}->param("modestmap.composer");
444 |    my $path_python = $self->{'cfg'}->param("modestmap.python");
445 |
446 |    my $provider = $self->{'cfg'}->param("modestmap.provider");
447 |
448 |    $provider =~ /^([_]+)$/;
449 |    my $short = lc($1);
450 |
451 |    my $acc = $self->mk_flickr_accuracy($short, $acc);
452 |    my $out = $self->mk_tempfile(".png");
453 |
454 |    my $h = $self->pinwin_map_dimensions("height");
455 |    my $w = $self->pinwin_map_dimensions("width");
456 |
457 |    my $cmd = "$path_python $path_composer -d $h $w -p $provider -
c $lat $lon $acc -o $out";
458 |    system($cmd);
459 |
460 |    if (($!) || (! -f $out)){
461 |        $self->log()->error("Failed to create modest map, $!");
462 |        return undef;
463 |    }
464 |
465 |    return $out;
466 |}
```

Ideally, I would like to write a "ws-compose.py" endpoint, to be run locally or on a remote server, so that users can simply point to something starting in `http://` in their config file. There's also the part where I haven't actually tested placing thumbnails for **any provider except**

Yahoo! <http://www.aaronland.info/weblog/2007/06/08/pynchonite/#net-flickr-geo>

~~I am going to read the paper, now, but I will try to push something out later today...~~

Net::Flickr::Geo.pm

0.3 <http://search.cpan.org/dist/Net-Flickr-Geo/>

2007-06-28T08:39:50-0700

FIREBAGEL!!!!!!

There's lots to write about **the trip to**

Europe <http://flickr.com/photos/straup/sets/72157600484834830/> but that will have to wait for a lazy morning, with more coffee. In the meantime, I've written Perl bindings for the **FireEagle location service** <http://fireeagle.research.yahoo.com/> , recently announced at the **HackDay** **UK** <http://del.icio.us/tag/hackdaylondon> event.

FireEagle is not a magic bullet but it's an especially good toe-hold in a world where any sort of automagic tracking of your location is either a pain in the ass or just plain creepy. Usually both. Speaking of which :

Does FireEagle keep track of my whereabouts? <http://fireeagle.research.yahoo.com/faq.php>

Fire Eagle only remembers your latest location updates, and does not keep your location history. If you allow other applications to read your location, these application may build a history of your locations. We have no control over that - and recommend that you choose your trusted applications wisely. If you are unsure about applications you sign up for, you can always revoke their permission to access your account.

That's a bit of an artful dodge but it's also the only way to do it, in the short-term, given the quicksand surrounding privacy and location.

So, that's it. You tell FireEagle where you *are*, to whatever level of granularity you are comfortable with. And then you can ask FireEagle where you most recently *were*. And more importantly, third-party applications can ask where you were on your behalf *at a more or less precise level of granularity*.

Then, you know, you do stuff with that information. Here's some example code that uses my stored location to generate (machine) tags via the handy

Geonames <http://www.geonames.org/> database :

```
Readonly::Scalar my $GEONAMES_API_SCHEME => "http";
Readonly::Scalar my $GEONAMES_API_HOST   => "ws.geonames.org";
Readonly::Scalar my $GEONAMES_API_SEARCH => "/search";

sub main {
    my %opts = ();
    getopts('c:', \%opts);

    my $fe = Net::FireEagle->new(%opts{'c'});
    $fe->update_location("San Francisco CA");
    my $res = $fe->query_location();

    my $locality = $res->findvalue("/ResultSet/Result/city");
    my $region   = $res->findvalue("/ResultSet/Result/state");
    my $co       = $res->findvalue("/ResultSet/Result/countrycode");

    my $uri = URI->new();
    $uri->scheme($GEONAMES_API_SCHEME);
    $uri->host($GEONAMES_API_HOST);
    $uri->path($GEONAMES_API_SEARCH);

    $uri->query_form(style => "full",
                    country => $co,
                    fcode => 'PPL',
                    maxRows => 1,
                    name => "$locality, $region");

    my $req = HTTP::Request->new(GET => $uri->as_string());
    my $res = $fe->request($req);

    my $xml = $fe->parse_response($res);
```

```
my $geocode = $xml->findvalue("/geonames/geoname/geonameId");

my @tags = ("geonames:locality=$geocode",
            "geo:locality=\"$locality\"",
            "geo:region=\"$region\"",
            "geo:country=$co",
            );

return \@tags;
}
```

Which returns :

```
$VAR1 = [
    'geonames:locality=5391959',
    'geo:locality="San Francisco"',
    'geo:region="California"',
    'geo:country=US'
];
```

Ladies and gentlemen,

Net::FireEagle.pm <http://search.cpan.org/dist/Net-FireEagle>

2007-06-28T08:39:50-0700

Net::Flickr::Geo



For as long as we've had the ability to send (Flickr) photos to printing services I've wanted to make my own maps.

Which is tricky because there are almost no non-web interfaces to do the sort of API magic you can do with any of the big name

(**online** <http://www.aaronland.info/talks/papernet/>)

mapping services. Those that do exist all use the same tiles, namely low-resolution versions designed for the web. There's also the aggressively boring part (and despite any appearance to the contrary, I am pretty lazy) of having to keep up with whatever magic is necessary to stitch a bunch of tiles into a single "background" image.

Mike <http://mike.teczno.com/> tells me that ModestMaps has just this sort of **compositing magic built-in** <http://modestmaps.mapstraction.com/svn/trunk/py/composite.py> so I look forward to finding the time to be proven wrong. Beyond that, there is the **Yahoo! Map Image API** <http://developer.yahoo.com/maps/rest/V1/mapImage.html> which is so close to being what I want (read: stupid-dumb easy) and yet so far away (read : no markers, maps that don't lend themselves to printing).

When we got **back from Europe last month** <http://flickr.com/photos/straup/sets/72157600202022198/>, I finally sat down to make *something* using the Yahoo! APIs even if it's the sort of thing that I look back and laugh at in a couple years.

This is what I got :

```
my %opts = ();
getopts('c:', \%opts);

my $cfg = Config::Simple->new($opts{'c'});
my $fl = Net::Flickr::Geo->new($cfg);

my @maps = $fl->mk_pinwin_maps_for_set('72157600321286227', 'upload');
```

This is how it works :

1. *For every geotagged photo in a set :*

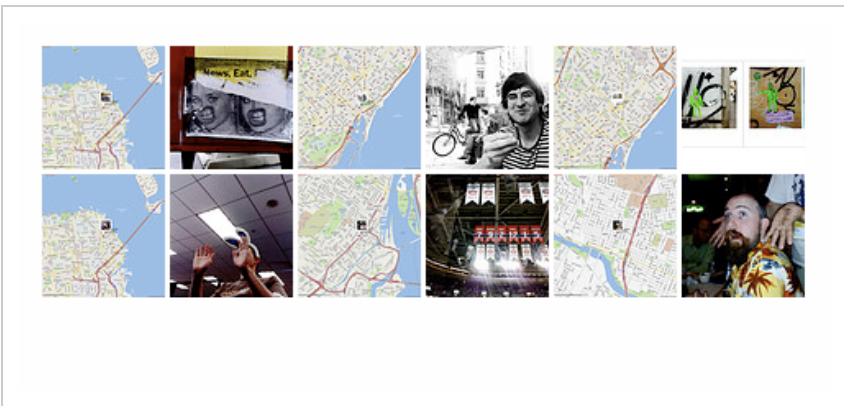
2. *Fetch the thumbnail and place it on a blank*

"pinwin" <http://l.yimg.com/www.flickr.com/image>


```
s/pop_up_pinwin.png "
```

3. *Fetch a map, using the Yahoo! Map Image API, corresponding to the photo's latitude and longitude and accuracy.*
4. *Place the newly created pinwin over the default map marker.*
5. *Upload the photo to Flickr, adding it to the same set as the original, updating the set's ordering such that the map always appears before the photo.*

In the end, your set looks something like this :



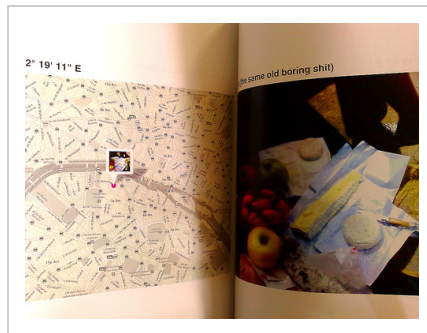
Which isn't that exciting except that it *is* enough to send off to **QOOP** <http://www.qoop.com/> for printing in a book (modulo boring details like a blank first image to ensure that maps are

always printed on even-numbered pages).

Which is kind of exciting because the maps help put the photo in context and make for better story-telling.

Barely related, at all, I've been reading **Against the Day** http://en.wikipedia.org/wiki/Against_the_Day which has its ups and downs but also includes this lovely passage :
"...pausing to gaze at ruined frescoes as if they were maps in which the parts worn away by time were the oceans..."

I guess the next step is to add the ability to place arbitrary markers on a map image. Then you could also query Flickr for other — yours or your contact's or everyone's— photos taken within the bounding box of the map and some limited window of time and see "from a distance" what else was happening when you took a photo.



Or begin to stitch together the map images to create maps of

arbitrary size — or, more specifically, big enough to fit any collection of photos at a given zoom level — to have printed as "posters". By which I mean : big sheets of paper which you can fold in to proper guides (even if the paper these things are printed still doesn't lend itself to that sort of thing...)

I'd also like to try all of this with other map providers, specifically the **Open Street**

Maps <http://www.openstreetmap.org/> stuff, where possible.

At a time when people are complaining about the lack of detail on maps I find myself wanting simpler, and more stylized, renderings; something to act as background music, or a soundtrack, to a photo rather than smothering it with an encyclopedic knowledge of urban minutiae.

So, anyway, there you go :

Net::Flickr::Geo.pm <http://www.aaronland.info/perl/net/flickr/geo>

If it's not already on the CPAN <http://search.cpan.org/~ascope/> by the time you read this, it will be soon.

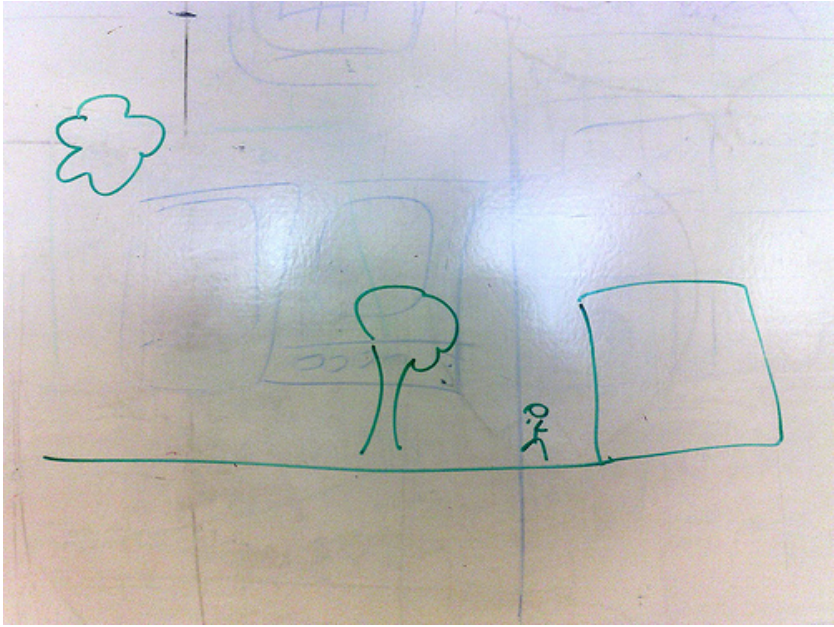
Meanwhile, back at the Ranch



<http://yahoorsearchberkeley.com/blog/2007/05/16/the-emerging-semantics-web-the-semantic-web-is-dead/>

The lovely **Dan Catt** <http://www.geobloggers.com/> and I will be in London this week, for Hack Day, **to bring you the machine tag love** http://developer.yahoo.net/hackday/2007/06/london_hack_day_schedule.html . I will also be around for the London "**24 Hours of Flickr**" <http://upcoming.yahoo.com/event/195695> " event as well as those in **Paris** <http://upcoming.yahoo.com/event/195694> and **Montréal** <http://upcoming.yahoo.com/event/195703> . Come say hello!

2007-06-02T05:44:20-0700



Songlines

I Hate Everything, The Musical

Je me souviens

I Hate Everything, The Musical

This is what I know.

- *The iPhone with its wireless-ness and ability to dock with any one of the many iPod speaker devices is not able to scan for an use shared iTunes libraries. **This is probably what the supposed "OS X iPods" are all about.*** http://daringfireball.net/2007/07/os_x_ipods
- *If you have a Firewire drive that you've formatted under OS X you need to be sure to **disable journaling in order to write to it in Ubuntu*** http://penguin.ubuntu.com/psych.gwdg.de/~ihrke/wiki/index.php/Installing_Ubuntu_on_iBook#Writing_on_HFS.2B_Journaled_file_system .
- ***Banshee*** <http://banshee-project.org/Distributions/Ubuntu> *can do recursive searches and export its library over **DAAP*** http://en.wikipedia.org/wiki/Digital_Audio_Access_Protocol *(the magic happy iTunes sharing protocol) and iTunes will see it but won't ever actually load it.*
- ***Rhythmbox*** <http://www.gnome.org/projects/rhythmbox/> *appears to be too stupid to recursively scan a directory looking for music files but if you go to the*

trouble of adding them one at a time, and then export your music library over DAAP iTunes will not only see it but also let you play it.

- *iTunes, meanwhile, will happily show you that your Rhythmbox library contains files that were encoded using Ogg Vorbis but it will not play them. Nope.*

- *Configuring **Sound***

Juicer <http://burtonini.com/blog/computers/sound-juicer> , the default Ubuntu CD ripping application, **to encode your music as MP3 files is actually pretty**

easy <http://ubuntuforums.org/showthread.php?t=295698> — *once you know what to do. Rather than just being a single button that performs all the necessary magic in background, though, you will need to copy and paste a suitable "pipeline" command and install stuff with the words "ugly" and "multi-verse" in their names. I kid you not.*

- *Which is mostly worth it because the pain of setting up MP3 encoding on Rhythmbox, and playing them through iTunes, is less than the part where it (Rhythmbox or Banshee or any of them, really) is woefully unable to stream anything to an Airport Express hub...**maybe*** <http://raop-play.sourceforge.net/> .

- *Over in left field,*
Canola <http://openbossa.indt.org.br/canola/> ,
the super magic Do What I Mean media player for the
Nokia (Internet)
tablets <http://www.aaronland.info/weblog/2006/01/09/hirst/#edfg2> *has no clue what to do with*
iTunes post the 7.0 release. They can see the DAAP
shares but that's about it. The good news is that
libraries shared from Rhythmbox can be viewed and
browsed. The bad news is that the only thing that
happens when you try to play a track is an error
message.
- *Likewise with*
Tangerine <http://www.snorp.net/log/tangerine> ,
an open-source cross-platform DAAP server, which
runs just great on OS X (you could continue to share
your own music with iTunes and then use Tangerine to
share the "Shared" folder, for instance) but has no
packages for Ubuntu.
- *Since Canola also supports something called*
uPnP http://en.wikipedia.org/wiki/Universal_Plug_and_Play *(DAAP for everyone but Apple is the*
easiest way to think about it) you can install a handy
application called
GMediaServer <http://www.gnu.org/software/gmediaserver/> *on the same box you're running your*
copy of Rhythmbox on. Then you can export all those

*music files using a different protocol but at least Canola (or **Media***

***Streamer** <http://downloads.maemo.org/product/mediastreamer/>) will play them.*

- *Unless they were ripped as Ogg Vorbis files. Or the file names start with numbers because that only **got fixed in version 0.10 of gmediastreamer** <http://www.gnu.org/software/gmediastreamer/NEWS> and the Ubuntu port is only up to version 0.9 and installing stuff by hand in Ubuntu always ends badly.*
- *When the N770 was first released, someone ported Rhythmbox but it was pretty flakey then and it doesn't seem to be maintained anymore. There are also **libraries for adding Ogg Vorbis support** https://garage.maemo.org/frs/?group_id=130&release_id=543 but they lack a proper installer and at least one of the dependencies won't even install under "OS 2006".*
- *Or you want either app to play more than one song at a time without human intervention. Or to even get to the end of the first one. No one, it seems, has any idea why this happens although there's been some speculation that it is caused by **incorrect metadata — the duration of a track — being written by iTunes** <http://lists.gnu.org/archive/html/gmed>*

`iaserver-devel/2006-07/msg00002.html` and thus causing uPnP clients to cry. Like babies.

- *This remains unclear to me since I ripped a CD as MP3 files using Rhythmbox and observed the same behaviour. Which means I can stop thinking about needing to re-encode all my MP3 files without VBR...*
- *On the other hand, the overall stability, packaging and ease of installing applications for the N700 (and I'll assume the N800 as well) has improved by several orders of magnitude in the last couple of years.*
- *And if you go to the trouble of **setting up Samba, or some other like system, to mount the above mentioned Firewire drive, plugged in to your Ubuntu machine, on your***
Mac `http://www.linuxquestions.org/linux/answers/Networking/File_sharing_with_OS_X_using_Ubuntu_5_04` *then you can keep using iTunes to rip CDs because, let's face it, it is fast and the others are not. The others are also, despite advertised, not able to detect new files in your "music" folder and import them into your library. But I guess you fight the battles you can win.*
- *Oh, and Series60 phones which all claim to have "support" for uPnP : That apparently does not include actually being to act as a client and, say, stream music off of another computer. There's just a lot of blather*

about "transferring" files from one device to another.

- *Speaking of which, you may be interested to know that not only is the recently released **Nokia Media***

Transfer application for OS

X <http://www.nokia.com/A4423134> brutally slow but it also copies all of the files you are synching from, presumably, iTunes to your phone in its own cache directory. The good news, I guess, is that most people aren't even aware that there are 4GB MicroSD cards never mind 8GB ones.

By which I mean to say : Rock on.

2007-07-15T15:07:00-0700

Je me souviens

This was a blog post I started in July of 2007 after a marathon two-week work-and-play trip to Paris, London and Montreal. I never finished the blog post and only discovered it again hiding in the comments, in 2013. Here is it, then, lightly edited for dramatic effect because this is the story I want to remember.

I want to remember eating alone at the ... *six years later I have no idea where I ate, that night.*

I want to remember forgetting that you have to dial 001 instead of 01 to reach an American number and because I was given a local number one-digit short and because the DHCP server at the hotel wouldn't give me an IP address I want to remember having no idea of **The**

Drama http://www.wired.com/business/2007/06/german_users_in/ that was unfolding, that night.

I want to remember walking back to the Opera from the 2nd in the rain wearing the leather shoes I bought on my first trip to Rome and which would still be wet in the morning. *Those shoes would finally see their end, a few years later, on an Oakland evening when I stepped off the BART on my way to a fancy restaurant and one of the heels simply fell off.*

I want to remember walking up rue Cler and texting Julie, who knows everything about Paris, to ask after a reliable Internet cafe.

I want to remember the soul-sucking chain, on rue St. Dominique, that Karl told us always had wireless proving to be as barren as the over-heated American coffee they sold to expats "living abroad".

I want to remember sitting on the cobblestones looking at the number of open networks with variations on the name "Pompidou" and wondering how many were simply phishing for passwords.

I want to remember deploying the site from the courtyard of the Pompidou.

I want to remember walking around the 14th missing all of Simon's phone calls before the SIM card I'd neglected to add money to in the 7th finally died.

I want to remember the other people who'd obviously seen **the article in the New York**

Times <http://www.nytimes.com/2007/04/29/travel/29Choice.html?pagewanted=all> and ordering lunch, not knowing whether Simon would ever arrive.

I want to remember that steak.

I want to remember the smell of all that cheese I'd carried around the city since morning opening up in the train as we passed under the English Channel.

2007-06-28T08:39:50-0700



Trying not to cut down the trees for an inability to see the forest

mmPDF.php 0.3

Small details, loosely crushing each other

[N]othing would be gained ...

S60 Stikkit 0.1

The Scribblenet

mmPDF.php 0.3

```

doap:Version      [
doap:revision      "0.3" ;
doap:created       "2007-08-14" ;
];

asc:changes       [
asc:addition       "Support for del:bookmark=geo.del.icio.us feeds" ;
asc:addition       "Support for KML files from platial.com" ;
asc:addition       "Basic support for ModestMaps ws-compose.py web service"
asc:addition       "Basic support for indexes at the end of a document" ;
asc:addition       "Optional max (items) parameter for 'add_map' method" ;
asc:addition       "Include Mike Migurski's JSON PHP class as part of the
distribution" ;
asc:addition       "Include qr PHP class as part of the distribution" ;
asc:addition       "Include pmPDF PHP class as part of the distribution" ;
asc:addition       "Include MachineTag PHP class as part of the distributi
on" ;
asc:addition       "Include Restobook PHP class as part of the distribution"
asc:comment        "Indexes are fugly; eventually this code will be moved ir
] .

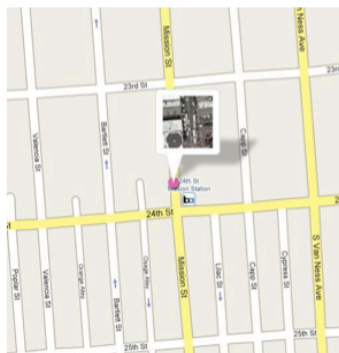
```

Enjoy: <http://aaronland.info/php/mmPDF/mmPDF-0.3.tar.gz>

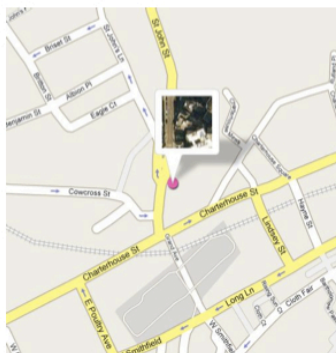
Actually, **enjoy version 0.31**,
more <http://aaronland.info/php/pmPDF/pmPDF-0.3.tar.gz> . Also,
pmPDF.php has been updated and released as **version**
0.5 <http://aaronland.info/php/pmPDF/pmPDF-0.5.tar.gz> .

2007-08-13T19:29:08-0700

Small details, loosely crushing each other



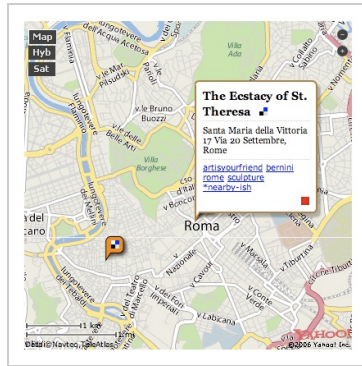
El Farolito
:yelp
"SGRmnarrNuVEsAjYdEoA0w";
:phone "4158264870";:address



St. John
:address "26 St. John St London
UK";

Paris <http://del.icio.us/straup/del:bookmark=geo+restobook+paris> , **Rome** <http://del.icio.us/straup/del:bookmark=geo+restobook+rome> , **London** <http://del.icio.us/straup/del:bookmark=geo+restobook+london> , **Montreal** <http://del.icio.us/straup/del:bookmark=geo+restobook+montreal> , **San Francisco** <http://del.icio.us/straup/del:bookmark=geo+restobook+sanfrancisco> and the place where the physical and digital world intersect /weblog/2007/07/28/trees/20070811_delmaps_pm_draft.pdf .

We'll take it as a given that no one has map tiles especially well-suited <http://www.termite.org/projects/maps/index.html> to piece-of-shit consumer printers, yet. In the meantime, I have added support for "del.icio.us maps" <http://www.aaronland.info/weblog/2007/07/28/trees/#delmaps> " to the mmpdf <http://www.aaronland.info/weblog/2007/05/21/playa/#mmpdf> PHP classes I wrote for my talk on the **Papernet** <http://www.aaronland.info/talks/papernet/> .



"mm" used to stand for

MyMaps <http://www.aaronland.info/weblog/2007/04/01/thick/#paperfs> .

Eventually there was an option to generate map images using

ModestMaps <http://www.modestmaps.com/> ; this is done using a bare-bones HTTP interface around the

compose.py <http://modestmaps.mapstraction.com/svn/trunk/py/compose.py>

script and will be checked in to the SVN trunk Real Soon Now was checked in as **Revision 339** to

/trunk/py <http://modestmaps.mapstraction.com/svn/trunk/py/> . Since then, I've also added support to suck places out of **Platial** <http://www.platial.com> and

del.icio.us http://www.aaronland.info/weblog/2007/07/28/trees/#delmaps_yelp ; again, expect a code release just as soon as I sort out the remaining details — the formatting on the addresses in the screenshot, above, for example — or at least write documentation with the **release of mmPDF.php**

0.3 http://www.aaronland.info/weblog/2007/07/28/trees/#mmpdf_03 . So, I guess "mm" just stands for

"mm" <http://www.flickr.com/photos/straup/541125822/> , now.

Oh, and remember the

barcodes? <http://www.aaronland.info/weblog/2007/02/17/platform/#barcode>

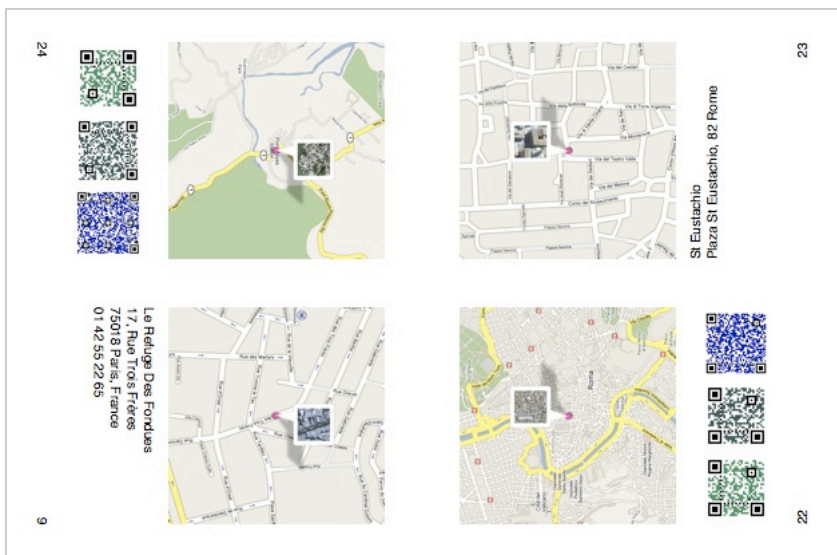
e Good, now imagine a world where del.icio.us had a proper mobile website for editing and posting addresses...I mean, links. See?

For extra points, consider all the chatter surrounding **Atom (the publishing part) and Google Earth** <http://del.icio.us/tag/atom+kml> ...

An update and a tease :

```
7 |         require("mmPDF.php");
8 |
9 |         $modestmaps = array('server' => 'http://127.0.0.1:9999/',
10 |                             'provider' => 'GOOGLE_ROAD',
11 |                             'marker' => 'YAHOO_AERIAL');
12 |
13 |         $qr = array('data' => './qr_data',
14 |                     'images' => './qr_images');
15 |
16 |         $args = array('folds' => 0,
17 |                       'source' => 'modestmaps',
18 |                       'modestmaps' => $modestmaps,
19 |                       'qr' => $qr);
20 |
21 |         $pdf = new mmPDF($args);
22 |
23 |         $pdf->SetFont('Helvetica');
24 |         $pdf->SetFontSize(10);
25 |
26 |         $pdf->add_map("chris heathcote's massive mymaps dump of yummy food in paris", 5);
27 |         $pdf->add_map("my geotagged del.icio.us bookmarks for coffee");
28 |
29 |         $pdf->Output();
```

At this point, I think that the only thing left to do is packaging and documentation in advance of the next code release, including an updated version of the del.icio.us maps web interface. I'm not sure what to do in the way of barcodes for MyMaps feeds so I may just leave that one for later. After that, I will add tag-based (and properly formatted) indexes to the long list of tiny details.



But for now, **here's a big PDF**
file /weblog/2007/07/28/trees/20070812_delmaps_pm_draft-2_mixed.pdf
 instead.

2007-08-12T14:05:50-0700

"[N]othing would be gained from pursuing the abstractions..."



Yelp released their API <http://www.yelp.com/developers> the other day so I added hooks to the del.icio.us/restobook maps <http://www.aaronland.info/weblog/2007/07/28/trees/#delmaps> project to pull in Yelp listings within a one-mile radius of any geocoded query. In the screenshot, above, the result for **1062 Valencia San Francisco CA** <http://flickr.com/photos/tags/ritualroasters/clusters/sanfrancisco-coffee-ritual/> is displayed in the top-left corner and nearby stuff, from Yelp, around it. This is good since it makes it easier to generate **restobook** <http://www.aaronland.info/weblog/2006/07/03/smoking/#restobook> style descriptions and generally be more precise (than the freakishly bad reverse-geocoding done by geonames) when tagging neighbourhood and city.

I will post an updated public version next week but in the meantime here's the part about the API that's not mentioned anywhere in the docs :

You can filter search queries by "category"; more goodness. However not only are those categories not listed anywhere in the documentation they are also *not* returned as part of the list of categories for an individual search result. For example, El Farolito returns "mexican" but is filtered on by "restaurant", by passing a `category` parameter. Anyway, if you sniff the URLs in the "Browse By Category" list on the **front page** <http://www.yelp.com/> you can find a useful list of terms to limit queries by.

Also, if you think this is all crazy-talk consider the recent announcement about **support for Microformats in Google**

maps <http://googlemapsapi.blogspot.com/2007/06/microformats-in-google-maps.html> .

2007-08-03T19:14:29-0700

S60 Stikkit 0.1

I always feel bad when **I don't say nice things about**

Stikkit <http://www.aaronland.info/weblog/2007/07/28/trees/#delmaps> .

I was poking around, yesterday afternoon, for a recipe for **Swedish ginger cookies** <http://del.icio.us/url/daa376688596bb045afaa46c8368f8ef> published a while back in the New York Times.

Eventually I found it on Chowhounds and, for safekeeping, promptly copy-and-pasted it in to Stikkit.

Then I thought about **printing out the recipe** <http://www.aaronland.info/weblog/2006/12/17/meat/#papernet> all properly formatted and nice-like.

Then I simply wrote the recipe down on an index card.

Then I thought about taking the index card — as in the list of ingredients — to the store.

Which got me thinking about printing again **which is really a trap door in to the abyss.** <http://www.aaronland.info/weblog/2006/03/16/terroir/>

Then I got a little depressed.

Then I pressed F12 (the magic key to display the extra handy **Stikkit Dashboard widget** <http://interblah.net/2007/4/3/stikkit-dashboard-widget>) a few times and thought : It would be good to have this on my phone. So I wrote an application, in Series60 Python, to just that.

Specifically, it does exactly three things : Lists your (recent) Stikkits; Displays a Stikkit; Allows you to create a new Stikkit. That's it (modulo tweaks to the posting interface because the Series60 Form widget blows).

Then I got depressed, again, because I mistakenly thought that the current release of S60 Python (1.40) was yet-another clusterfuck of undebuggable fatal errors. Eventually, I realized that I had simply forgotten to *"import generators from the future"*. Then I got depressed again.

But, at least it works : **(S60) Stikkit.SIS**

0.1 http://www.aaronland.info/bin/s60_stikkitapp/

(Where "works" means you'll need to get an **API key** <http://www.stikkit.com/api> .)

2007-07-30T09:18:49-0700

The Scribblenet

Executive Summary : **Data, Not**

Answers <http://www.aaronland.info/talks/scribblenet/>

A little over a year ago, I came back from

Helsinki <http://flickr.com/photos/straup/sets/72157594187100815/> and resolved to finally get around to learning **Nokia's Series60**

Python <http://opensource.nokia.com/projects/pythonfors60/> . This is what I wrote, about a month later, when I posted the first release of an application called **restobook** <http://www.aaronland.info/weblog/2006/07/03/smoking/#restobook>
ok :

But I'd managed to do enough hacking to get a feel for how things worked and I started to think about what else I could suck in to the address book. Restaurants seemed like an obvious choice and keeping restaurants listed in del.icio.us also seemed like a good idea. I suppose there are legitimate privacy concerns but to my measure they aren't a big deal and don't outweigh the benefits : Ready-access from any web browser, not having to deal with synching and just otherwise sharing with the community. For example, I spent a lot of time poking around /helsinki and /helsinki+restaurant before I left for Finland and it was very useful.

In March, of this year, **I sat on a stage and told people not to despair (too much)** <http://aaronland.info/talks/sxsw07/> since pretty soon all applications would be written as glorified web pages. I was the Doom & Gloom guy on a panel about the current state of mapping, having by then spent more time than is healthy trying to build tools using S60 Python. The short version is that the Symbian 9 security policy, which dictates the functionality an application can use, is dumb and lazy and cumbersome. **In effect it makes doing anything so painful as to be impossible.** http://blogs.s60.com/tommi/2006/10/symbian_signed_should_it_be_ch.html#comment-12615



All the talk about whether Perl, Python, PHP or Java will assume the mantle of "lingua franca" on the Internets has become moot since JavaScript already is. The recent announcement that Mozilla's next JavaScript engine, Tamarin, will **also be a container for functionality written**

in http://weblogs.mozillazine.org/roadmap/archives/2007/07/new_projects.html Python and Ruby (and, one assumes, beyond) is proof that **JavaScript is the new Parrot** <http://www.parrotcode.org/> .

In May, I sat in the audience while a nice chap from the Nokia Research Center (NRC) demonstrated an address book application running on the phone and written in HTML + JavaScript, specifically a JavaScript wrapper around the device's core Series60 C++ APIs. At the end of his presentation he sheepishly admitted that they hadn't really bothered to sort out the problem of how to deal with the security restrictions. This was a "research project", after all, so they fudged it which is **a curious variation on the whole idea of**

vapourware <http://xtech.expectnation.com/event/1/public/schedule/detail/210> .

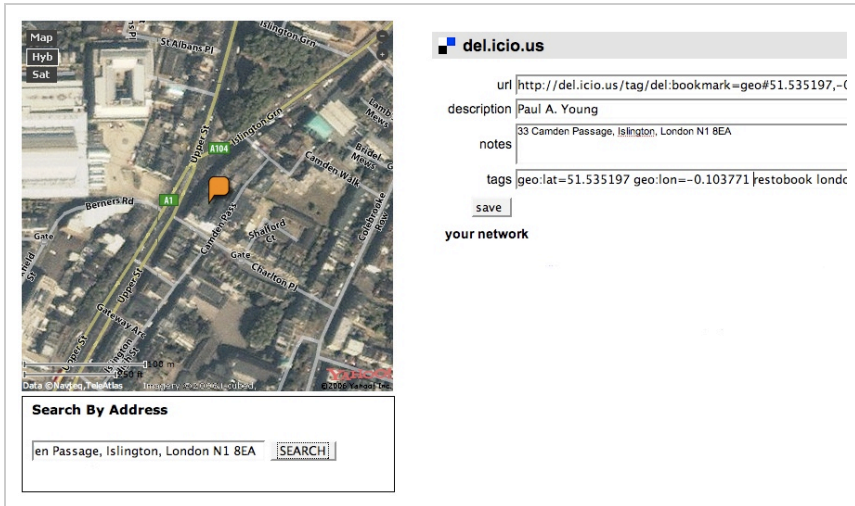
As I started grouching to the person sitting next to me another Nokia employee piped up and demanded whether I thought they should be releasing "insecure" devices on the market. This is a mostly spurious argument because the answer is obviously no but that is not a justification for the remarkably bad design

and implementation of the Symbian 9 security "platform" which mandates that all applications be signed *by* Symbian which, at the end of the day, means that all developers are suddenly held hostage.

The mobile community is especially curious in this respect, partly due to the history of cellular phones : They operate under the assumption that you should be grateful you are allowed to build anything in their magic gated communities. It remains a mystery whether they are too dumb, or just willingly ignorant, to see that "mobile devices" have become nothing more special than really small computers with any number of arbitrary network connections. And like any other computer, or "platform", its value is directly tied to the willingness of and the ease with which developers can write tools on top of it.

Whereupon I counter all arguments to the contrary with The Sword of Flawless Victory called "Facebook".

In June, Steve Jobs announced that developers could write applications for the iPhone as "web applications", with all the usual blather about security and mobile devices. Everyone groaned. As if a company that's been selling operating systems for 20+ years can't figure out how to create an environment where developers and users can play nicely, and *safely* with each other.



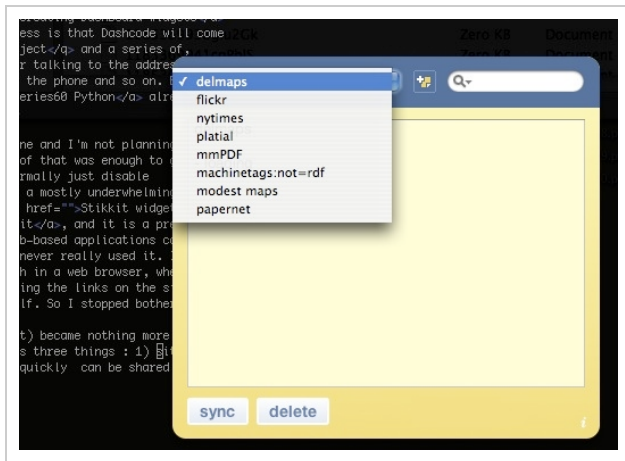
It is worth noting that Apple already has a way for users to develop tools as "web applications". They are called **Dashboard**

widgets <http://www.dashboardwidgets.com/> and while they are still mostly just toys they are also just this : HTML + JavaScript + custom extensions to wrap low-level Objective C functionality. I can only guess that the security "model" here is the sum total of people (read: users) not being complete morons and the barrier to entry for any sort of programming that requires "compiling".

I could easily be wrong but it's hard not to look at the tea leaves and think that applications for the iPhone — because no one can really believe in a world where it continues to exist as a closed device — will be written as widgets. The current thinking is that when the next version of OS X (10.5) is released it will come with all manner of sexy integration with the iPhone. Which is also when **Dashcode, the development environment for creating Dashboard**

widgets <http://developer.apple.com/tools/dashcode/> is set to be released. My guess is that Dashcode will come with a special iPhone "project" and a series of, probably restricted, APIs for talking to the address book, and the wireless network and the phone **and so on** <http://www.ditchnet.org/wp/2007/03/01/lets-try-this-again-canvasuml/> . Basically everything that S60 Python already does. And probably more.

Soon the winter months will come and the Finns will not see each other shielded by their masks of frozen tears.



I still don't own an iPhone and I'm not planning to buy one, any time soon, but all of that was enough to get me to revisit Dashboard which I normally just disable entirely. It continues to be a mostly underwhelming experience except for the **Stikkit widget** <http://interblah.net/2007/4/3/stikkit-dashboard-widget> . I really like **Stikkit** <http://www.stikkit.com/> , and it is a pretty fantastic example of what web-based applications can be, but the brutal truth is : I never really used it. It's just too much trouble to deal with in a web browser, whether it's typing out the URLs or clicking the links on the site or waiting for the browser itself. So I stopped bothering.

That is, until it (Stikkit) became nothing more than a really dumb notepad that does three things : 1) sits patiently in the background; 2) loads quickly 3) can be shared between disparate machines. You know, everything that the web is supposed to be. But isn't. There's no capital-R revelation in that one other than to slap the "Mozilla will save us" people in the face with a soggy fish.

Meanwhile, the NRC's quirky **mobile web server** <http://research.nokia.com/research/projects/mobile-web-server/> has been re-born as a **real-live and fully branded**

application <https://mymobilesite.net/> . Which is interesting because under the cover this thing is just Apache (read: HTML) running Python scripts (read: JavaScript (all grown up) with extensions to the core C++ APIs). *With permissions to do all the stuff that you can't do as a mere mortal (read: developer).*

In May, **I also stood in front of a crowd and told them about something called the Papernet** <http://www.aaronland.info/talks/papernet/> citing, in particular, Google's MyMaps as an example of the place where the physical and digital world

intersect <http://www.aaronland.info/weblog/2007/04/01/thick/#paperfs> . To make the point, I wrote

mmPDF.php <http://www.aaronland.info/weblog/2007/05/21/playa/#mmpdf> which will take a KML feed and create a nicely formatted **PocketMod**

booklet <http://www.aaronland.info/weblog/2007/04/01/thick/#pmpdf03> .

And despite the heavy precense of Google weenies, at the conference, telling us about all the magic GData APIs there are still no endpoints for doing anything with MyMaps, programtically.

Which is a shame because MyMaps is a really good candidate for a stripped down Dashboard style "local" application. Something that would allow a user to quickly enter unstructured data (geocoding and tags), handing off to the computer to do the rest work, storing everything in a standard format (latitude and longitude), and later generating a faceted version (all the restaurants tagged "french" in San Francisco).



So I wrote my own. Using machine tags and del.icio.us as the datastore.

By which I mean, it is nothing more than a web page that will load a **(Yahoo!) map** `http://developer.yahoo.com/maps/ajax/index.html` and provide a bare-bones interface for geocoding an address, placing a marker on the location that the geocoder thinks is the best match. The marker can be dragged around the map to correct the location. When you double-click the marker, it loads the **del.icio.us posting interface** `http://del.icio.us/post?title=The Scribblenet&url=http://aaronland.info/weblog/2007/07/28/trees#delmaps` in an iframe passing the geocoded address and the latitude and longitude data along with it. In addition it sends some extra "marker" tags, notably `del:bookmark=geo`.

The part where the tool uses an iframe raises at least one issue : It's hard to capture, let alone evaluate, events that occur in the iframe. It would be nice to be able to convert the previously dynamic marker into a fixed one when the save (to del.icio.us) button is pressed but I haven't figured that one out yet... The part where all the login/authentication nonsense is left for del.icio.us to deal with is almost

worth it. Almost.

To find all of the things you've assigned geodata to you simply need to start querying for stuff tagged with

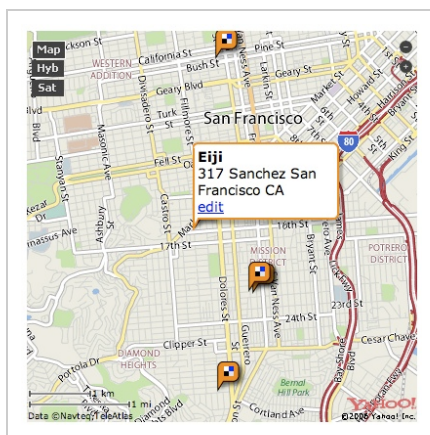
del:bookmark=geo <http://del.icio.us/tag/del:bookmark=geo> . Or

del:bookmark=geo+restobook <http://del.icio.us/tag/del:bookmark=geo+restobook> . And so on. More likely you'd do something like

del:bookmark=geo+paris <http://del.icio.us/tag/del:bookmark=geo+paris> .

If Joshua didn't hate machine tags so much, you might be able to prevail on him for the ability to say : What are the distinct values for, say, `geo:locality=` tags? That way you could create a nicer interface for browsing the available data and, potentially, make the application a little zippier. Come to think of it, it would be awesome if everyone who offered machine tags did that.

Um. Yeah. Did I mention that I did a whole talk on machine tags in June? <http://www.aaronland.info/talks/hackdayuk07/> *Anyway, moving right along...*



In principle, you can easily fetch and display those posts on any old web-based map by asking del.icio.us for results output as JSON. In principle because the JSON feeds are restricted to only public bookmarks and because the JSON endpoints are limited to individual users. (It also seems that you can only query on a

maximum of two tags.) Which kind of sucks because I would like to see what *other* people say for `helsinki+againagain`. But these are "design" decisions and not anything inherently undoable. There is nothing to stop you from setting up a proxy server that performs authenticated calls to the **entire del.icio.us**

API `http://del.icio.us/help/api/` and sending back JSON.

In its current iteration, it also does reverse geocoding using the **Geonames** `http://geonames.org/` web service, plucking out the names of each place and assigning them as tags. So far, the anecdotal evidence suggests this should be optional; anyone whose ever spent any time in Montréal know that Longueuil is not the same as Little Burgundy. Other caveats include : It is still kind of ugly and I won't be surprised if there are still some gotchas with the draggable markers.

On the upside : It actually works, is easy and really simple to "install". And it will Just Work with

restobook.py `http://aaronland.info/python/restobook/` (assuming I ever get around to porting it to Series60 3rd Edition).

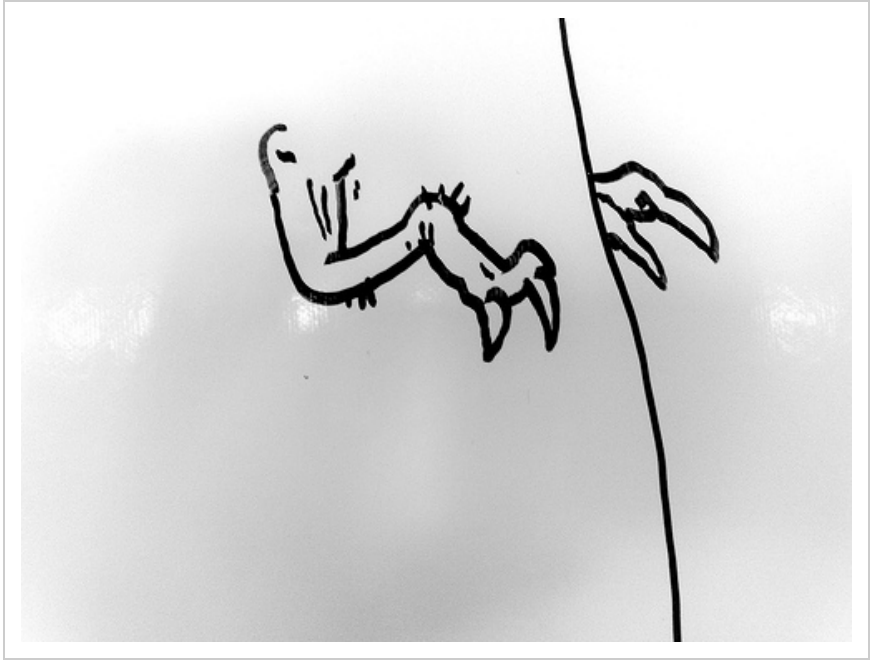
One of the next steps will be to update **mmPDF.php** `http://aaronland.info/php/mmPDF/` to fetch a list of data points from del.icio.us. I finally got around to writing the **web service interface to the ModestMaps compose**

tool `http://www.aaronland.info/weblog/2007/06/08/pynthonite/#nfg_mm` that I threatened about, a while back, so I'd like to try and bundle all of that up in **one big release** `http://flickr.com/photos/straup/899236402/` . As usual, we'll see.

And since I seem to be spending lots of time in meetings these days that should offer plenty of chances to clean up the interface. Do not pretend that I think it is pretty but right now my fingers are still bleeding like a thousand callbacks from having to work through all the nonsens... I mean *fun* that is programming in JavaScript. Suggestions and cluebats are welcome.

So yeah, my maps. **My tasty del.cio.us maps.** `/weblog/2007/07/28/trees/delmaps/`

2007-07-28T13:57:55-0700



**Is that an address in your
pants?**

The Taste of Doom

The Taste of Doom

I woke up this morning to discover that **the Microformat people have turned their attention to recipes** <http://microformats.org/wiki/recipe-formats> .

Despite my overall reservations about Microformats, I think this a *good thing*. Anything that makes it is easier to get stuff *out* of the web is okay by me. But I also tend to view stuff like Microformats and the newly minted **HTML5** <http://www.ibm.com/developerworks/xml/library/x-html5/> as, basically, an effort to trick people into turning the web (read : HTML) in to **DocBook** <http://www.docbook.org/> .

That is not necessarily a bad thing. I happen to love DocBook but I also know, even in its simplest forms, it can be a chore to write. Microformats, doubly so because (at least until HTML5 is deployed and tries to pretend everything in the world is a blog post) they rely on nested tags with semantic attributes rather than self-identifying element names.

Even then, and having actually written (read : tried to use) the spec for Eatdrinkfeelgood 1.0, well-defined and considered elements names are not a panacea. It's just too much of a pain in the ass to want to write anything down. This is not a universal rule but I will stake the claim that it is true for recipes.

The response I usually get from any kind of markup enthusiast — and let's not discriminate, this is equally true of Microformat weenies as it is of RDF nerds or Atom wonks — is

that "**a machine will write it for**

you <http://food.yahoo.com/> ". Which is, I think, the part where people reach the fork in the road. It's not that anyone is (more) right or wrong, it's just that they disagree on fundamentals. To an outsider, it looks like Google believes *everything* will happen online and the Microformats crowd expects that HTML will become (read : already is) the Lingua Franca of ... everything.

I don't. Or, rather, I'd prefer something a little more flexible.

Anyway, since **the article about the design of Eatdrinkfeelgood**

1.0 <http://www.xml.com/pub/a/2005/02/16/edfg.html> was linked to and since so much of my thinking has evolved since then I sort of feel the need to make it clear that : Everything I said is still true. Except for the parts that have changed.

To whit :

- *January 2006 — Everything I said I wouldn't do* <http://www.aaronland.info/weblog/2006/01/09/hirst/#edfg2> (or *EDFG 2.0*)
- *March 2006 — Next Steps* <http://www.aaronland.info/weblog/2006/03/16/terroir/#nextsteps>

- *March 2006 — <you> :a "what you eat"*
<http://www.aaronland.info/weblog/2006/03/16/terroir/#erdfg>
- *April 2006 — e(r)dfg-writer* <http://www.aaronland.info/weblog/2006/04/19/feedme/#erdfg-writer>
- *September 2006 — The Illusion of Easy* <http://www.aaronland.info/weblog/2006/09/28/bake/#easy>
- *December 2006 — The Papernet* <http://www.aaronland.info/weblog/2006/12/17/meat/#paper.net>
- *February 2007 — Writing on the Wall* <http://www.aaronland.info/weblog/2007/02/17/platform/#wall>

At which point, the discussion starts to move further and further away from actual recipes but where everything that comes after it is a direct result of trying to figure out what to do about Eatdrinkfeelgood.

In March, of this year, I did a presentation called **The Scribblenet** <http://aaronland.info/talks/scribblenet/> where I used addresses, wine listings and recipes to illustrate the

problem of "walking the line between making it easy enough for people to bother putting data in to a system and still useful enough to make it worth the trouble of getting it out". By which I mean, the first two are only as manageable as the third isn't.

DOOM. DOOM.

DOOM. http://www.aaronland.info/weblog/2007/07/28/trees/#s60_stikkitapp

Oh, and there is the small part where I forgot to renew the registration for eatdrinkfeelgood.org so all the stuff that used to live there, uh, doesn't. (Always online all the time, right?) I now own

eatdrinkfeelgood.info <http://eatdrinkfeelgood.info/> but I still need actually get the DNS working properly. I feel reasonably confident that I can do that before anyone figures out the larger problem of how to actually markup a recipe.

I briefly considered just adding that list to the Microformats wiki itself, but that seemed like bad-form at best and spamming at worst.

Profit!

2007-08-22T08:00:58-0700



**“Aware of only one voice from
above”**

I just want to share my social network... with...
What the fuck are graduate students doing these...
Release notes are boring

**I just want to share my social
network... with myself**

Recently, someone (**Kellan** <http://www.laughingmeme.org/> , I think) said to me:

*You know I am right there with you with your "**Your Address Book is the Center of your Social Network**"* <http://www.aaronland.info/weblog/2006/07/03/smoking/#restobook>
*" Mantra... Surprised no one has created a plugin for Apple's. It would be *A Good Start TM**



And I said:

*Back during the days of the **Job Search Before***
Flickr <http://www.aaronland.info/weblog/2004/10/22/5556/> *I was slowly trying to teach myself Cocoa for just this reason; essentially the germ that has grown into "**restobook**"* <http://aaronland.info/python/restobook/> *" (and **del.icio.us***
maps http://www.aaronland.info/weblog/2007/08/24/aware/#delmaps_02 *). The problem is that there is no way to modify the default "panel" for an **addressbook***
entry http://developer.apple.com/documentation/AppleApplications/AddressBook-date.html#//apple_ref/doc/uid/TP30000440-TP30000418-TP30000442 *. You can only add overlays, which made the whole thing kind of suck.*

So, I am reduced to writing S60

apps. <http://www.aaronland.info/weblog/2007/01/03/itemz/>

*Now that I have a working developer's certificate I am revisiting things like the original **fl(ickr)addressbook** and the*

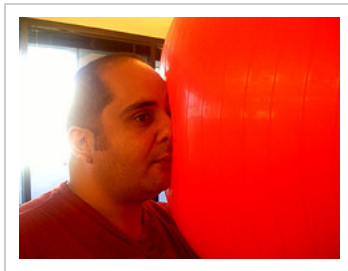
nwtracker <http://www.aaronland.info/weblog/2006/08/04/zebra/#watching> and poking at the

"landmarks" <http://contextwatcher.lab.telin.nl/ContextWatcherPortal/software/landmarks.zip> " (the maps app) database. The latter is actually sort of interesting since you can populate it on the fly. It is not so interesting because it doesn't have any sort of hooks into the address book beyond abusing existing fields.

*In *theory* you could run a copy of*

raccoon <http://opensource.nokia.com/projects/mobile-web-server/> (web server) on the device and assign the www link fields in both the contacts and landmarks database to point to each other and ... uh ...

synergize <http://www.aaronland.info/weblog/2006/08/04/zebra/#upcalenda>
r ?



Which is kind of a **round-about way of**

saying <http://schulzeandwebb.com/blog/2007/09/09/the-experience-stack-at-dconstruct-2007/> that I wrote a Perl module on top of Christophe Beauregard's **Flickr::Upload.pm** <http://search.cpan.org/dist/Flickr-upload/> library to fetch your current location, according to the **Dopplr API** <http://blog.dopplr.com/index.php/2007/09/03/announcing-the-first-release-of-dopplr-api/>, and to automatically add tags, machinetags and

geotags accordingly.

For example, if you uploaded a photo while you were travelling in **Beijing** <http://www.dopplr.com/place/cn/beijing> the following extra bits of information would be added:

- The tag **Beijing** <http://www.flickr.com/photos/tags/beijing> would be added to your photo.
- The machine tag **geonames:locality=1816670** <http://www.flickr.com/photos/tags/geonames:locality=1816670> would be added to your photo because Dopplr uses **Geonames** <http://blog.dopplr.com/index.php/2007/08/24/dopplr-gets-a-gazetteer-upgrade/> for their geocoding and that's **their unique identifier for Beijing** <http://ws.geonames.org/hierarchy?geonameId=1816670> .
- If you allowed it to be, the machine tag **dopplr:trip=9999999** (read : whatever your actual trip ID is) would be added to your photo. I wonder if you'll ever be able to share trips in Dopplr...
- Your photo would be **geotagged** <http://blog.flickr.com/en/2006/08/29/geotagging-one-day-later/> with a latitude of 39.9289 and a longitude of 116.388 and an accuracy of 11, or "city" level.

In the same vein, **YBR's ZoneTag** <http://zonetag.research.yahoo.com/> has been using **Upcoming** <http://www.flickr.com/groups/zonetagusers/discuss/72157594258417964/> , to infer location, since last year and the soon-to-be-public **FireEagle** <http://www.aaronland.info/weblog/2007/06/08/pynchonite/#firebagel> is the same idea bundled up as a discreet service. So, if we **can't ever figure out who is on first** <http://bradfitz.com/social-graph-problem/> we can at least litter the Intercloud with little fortune cookies made of magic words.

Tagged with [dconstruct](#) [dconstruct07](#) [barcamp](#) [barcampbrighton](#) [barcampbrighton07](#)

Illustrations

Thanks to the magic of [machine tags](#), you can illustrate this post by tagging a picture on [Flickr](#) with: `ladactio:post=1341`



Anyway,

Flickr::Upload::Dopplr.pm <http://aaronland.info/perl/flickr/upload/dopp>

1r/

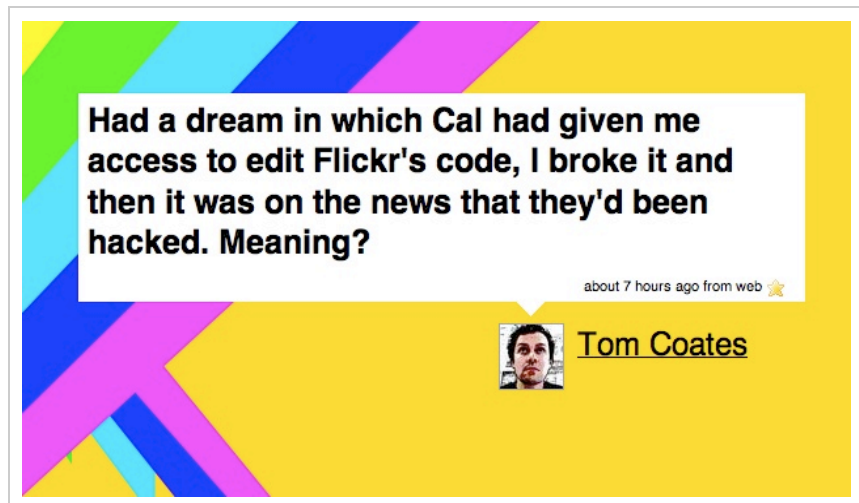
2007-09-09T14:10:21-0700

**What the fuck are graduate students
doing these days?**

22:53:47 M: why so many?
 22:53:53 M: all combinations?
 22:54:00 A: yes
 22:54:03 A: leftmost-iness
 22:54:33 M: I don't see much of a way around that
 22:54:38 A: because you will want to GROUP BY document_id
 22:54:57 A: in order to do LIMITS without totally fucking up pagination
 22:55:41 A: that was the first lesson of flickr for me : all bets are off the moment you decide the
 22:56:01 M: bad decision ;)
 22:57:15 A: but really, it's all INT-y indexes except for the string_value stuff so...all things cc
 22:58:09 A: but it still begs the question : what the fuck are graduate students doing these days?
 22:58:49 A: "I hold a doctorate from the University of DWIM"
 22:59:41 M: they're all working on the semantic web
 22:59:47 M: and polygon counts
 22:59:52 M: and natural language web search
 23:00:58 A: "the cloud will save us"
 23:02:14 A: "how many polygons does a cloud have"
 23:02:48 A: "would you like to buy a fog machine"

In October I am going to do a talk on machine tags, at the **Semantic Web Strategies**

conference <http://www.semanticwebstrategies.com/conference/sessionsbyday.php#A2> , in San Jose. Because talk is cheap, I always try to have something that looks like working code to make my case. I could, of course, **simply point to Flickr** <http://www.flickr.com/groups/api/discuss/72157594497877875/> .



But here's the thing : We have to architect things, including machine tags, differently than you. That is not a value judgement or posturing. It's just true. We have over one and a quarter billion photos.

And databases suck (which I'll get to in a minute).

There's lot of space between nothing and a billion and that's always been the place where I've hoped that people **would play with machine tags** <http://del.icio.us/tag/machinetags> . Ideally something you could plug into Wordpress or some other publishing system. Nothing fancy. Just a tool to index machine tags and provide a search interface so that you could link up disparate data sources.

In vanilla web-services-speak, you might say :

- *mt.namespaces.list(predicate="", value=")*
- *mt.predicates.list(namespace="", value=")*
- *mt.values.list(namespace="", predicate=")*
- *mt.documents.search(namespace="", predicate="", value=")*
- *mt.documents.search_by_range(range, namespace="", predicate=")*
- *mt.document.add(uri, tags)*
- *mt.document.remove(uri)*
- *mt.document.tags(uri)*

So I wrote one.

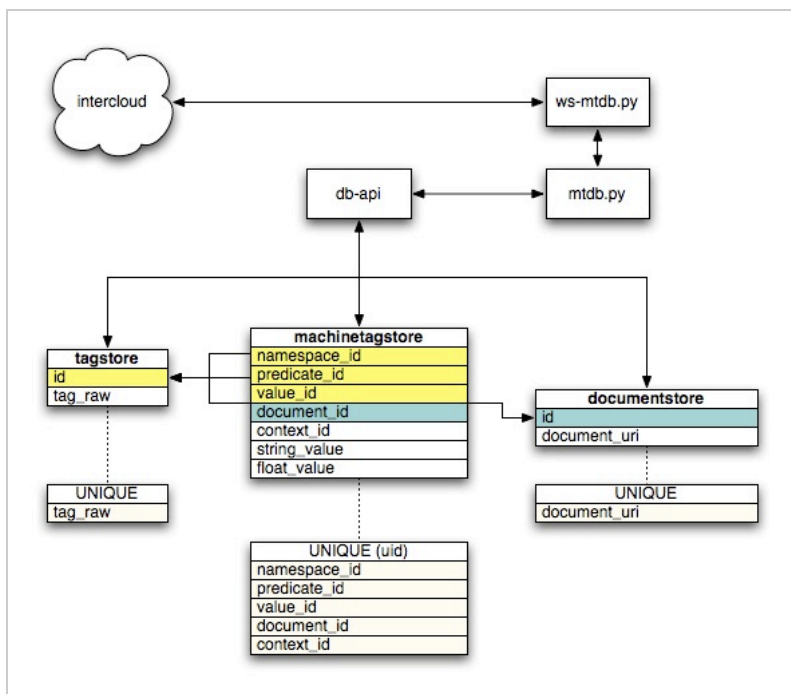
Specifically, I wrote **a really simple stand-alone machine tag store** <http://www.aaronland.info/python/mtdb/> that does not have ponies and will, hopefully, act as a spark for someone(s) to take it further.

It could be that I am the only person out there who really likes machine tags. I prefer, though, to think that part of the reason no one else has done this is that

while machine tags are pretty simple, conceptually, by the time you start thinking about storing and querying them it begins to get ugly and complicated.

Typically, when you start to capital-T "think" about the problem you find yourself saying things like : *"Well, I could install Lucene"* or *"What about a triple store?"*. While there's nothing wrong with these approaches they're a little like telling the person who wants a glass of milk that they need to buy a cow.

I opted for making **Python 2.5** <http://www.python.org/> the only requirement, and to use the built-in **sqlite3** <http://docs.python.org/lib/module-sqlite3.html> database magic.



Here's the rub : There are another **22 indexes** on the machinetagstore table, in addition to what is displayed in that diagram. By any measure that is too many, even if they are all mostly just collections of the integers whose order has been re-

arranged.

You could use a proper full-text search-engine to do this and not worry (as much) about indexing but then, more than likely, you won't be able to do range queries on your machine tag values. If you don't think you'll ever want to find stuff where, for example, the temperature was between -5 and 28 degrees Celcius then you could build something **using a custom Namazu**

filter <http://www.aaronland.info/weblog/2007/06/08/pynchonite/#nmzfilter>
r in time for dinner.

That said, range queries are easily the biggest feature request people have for machine tags on Flickr.

If you use a proper relational database then you get ranges but at the cost of "leftmost-iness". Specifically, to quote the **sqlite**
documentation <http://www.sqlite.org/optoverview.html> : *It is not necessary for every column of an index to appear in a WHERE clause term in order for that index to be used. But there can not be gaps in the columns of the index that are used.* This is a problem native to most (all?) SQL databases and the net result for machine tags is that unless you enforce a strict ordering of query parameters, thereby limiting the search-iness of your data, **you end up with indexes to match all the various**

combinations <http://www.phildawes.net/blog/2004/10/14/optimising-mysql-tables-for-rdf-store/> :

- *All the distinct namespaces*
- *All the distinct values, where the predicate is "subject"*
- *All the distinct documents, where the namespace is "dc" and the value is "new york"*
- *And so on...*

Where "distinct documents" in a database context best means the ability to

group results (read : documents) as part of the actual query so that you can reliably define an offset and limit to the number of results you return at once. Given that this isn't meant to scale to the moon, in the context of a home-user, or tinkerer, you could probably safely return all the results for a query and distinct-ify them in memory (read : programming language) but the issues are still the same.

All of this is made worse if you want to range and/or full-text indexing — or in my happy magic world a "context id", which is just an arbitrary numeric identifier that users can associate with a document — because **you just have to add that many more indexes** <http://gearon.blogspot.com/2004/05/addendum-i-noticed-someone-referring.html> . Did I mention that people really seem to want to be able to do range queries?

Or you can brush up on your

Java <http://www.russellbeattie.com/blog/java-needs-an-overhaul> and use Lucene which I'm told is smart enough to do (a convincing imitation) of range queries. Even then, though, you have to abuse the underlying model since you can't store the pieces of a machine tag as attributes. Or at least not more than one machine tag per document.

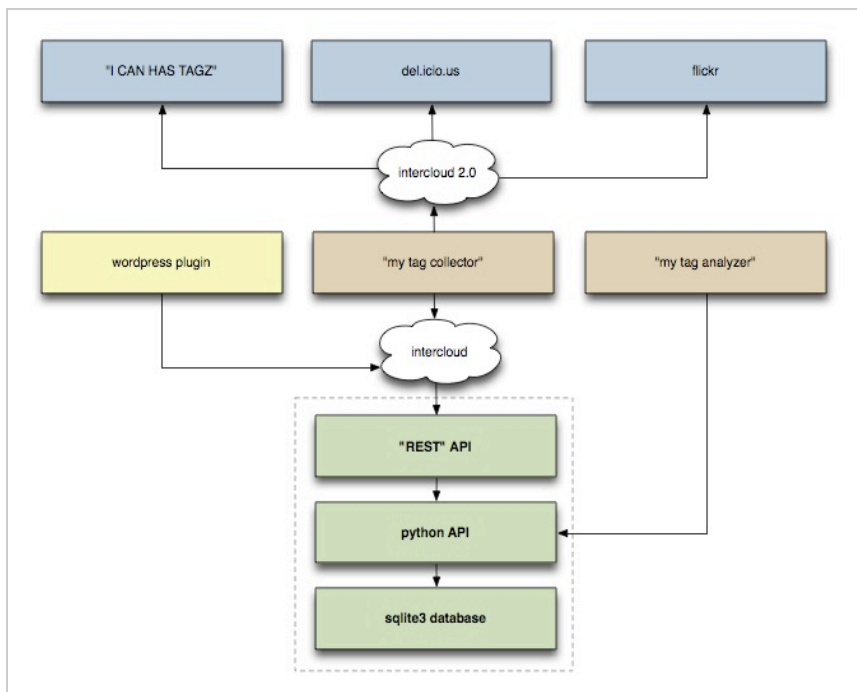
I would also like nothing more than to proven wrong about this because it's all a bloody nuisance. But I don't think I am.

So.

Really simple to install. And use.

Tinkeroserability.

How about we just worry about all those indexes if and when they become a problem?



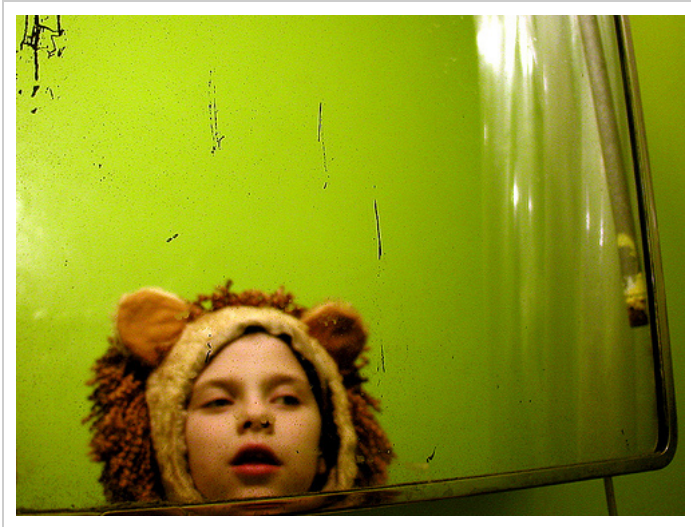
Out of the box mtdb (machinetag database) comes with a **standard Python interface** <http://aaronland.info/python/mtdb/README> for storing and retrieving machine tags associated with a document (read : URL) and a **bare-bones HTTP interface** http://aaronland.info/python/mtdb/README_WS_MTDB , that squirts out XML and JSON, for playing with the database from another language. Say, JavaScript.

The HTTP interface is nothing more than a proof-of-concept; a **Python script doubling as a web server** <http://fragments.turtlemeat.com/pythonwebserver.php> and meant to be run on a personal computer or trusted server. Understand that it has *no authentication or authorization mechanisms and its input validation is shockingly naive* so you should not expose it publicly without suitable modifications.

Improvements aside <http://www.python.org/dev/peps/pep-0333/> , it is also something that a person using another another publishing system could write

(or install) a plugin to use, with almost no additional overhead besides, like, the Interweb : 1) send an HTTP request 2) receive chunks of data formatted as *whatever* 3) profit!

And it implements all those methods `mt.*` API methods I described above.



Ladies and gentlemen, **mtdb.py**

0.1 <http://aaronland.info/python/mtdb/mtdb-0.1.tar.gz>

You will also need to install the *machinetag* <http://www.aaronland.info/python/machinetag/> and *simplejson* <http://cheeseshop.python.org/pypi/simplejson> Python libraries as dependencies.

2007-08-26T20:51:55-0700

Release notes are boring



Like most point releases, especially in the early stages when you're going from versions 0.1 to 0.2, it can be difficult to keep sounding the thunder. But there you go : **delicio.us**

maps <http://www.aaronland.info/weblog/2007/07/28/trees/#delmaps> has blessed as **version**

0.2 <http://www.aaronland.info/www/deliciousmaps/deliciousmaps-0.2.tar.gz> (complete with a tarball and everything!) and given a **more-better permanent home on the**

Interweb <http://www.aaronland.info/www/deliciousmaps/> .

The exercise with My Maps reinforced my idea that maps are, metaphorically at least, allelopathic. "The inhibition of growth in one species of plants by chemicals produced by another species" (source). So existing maps (a species?) poison (limit, prevent) a diversity of potential maps (yes I know biological metaphors for social phenomena can be dangerous). This is a particular problem with creative mapping tools (like My Maps) aimed at the general public who have seen mostly maps of the Google/Yahoo!/MapQuest species. What if you could map anything and you just mapped what is on typical maps?

John Krygier <http://www.cartoblog.com/allelopathic-maps-google-maps-my-maps/>

With that in mind, here's the list of things that have been added or changed in version 0.2 :

Magic Words

Restobook

To call "restobook" magic is being generous since it is just a collection bastardized machine tags whose syntax is a direct result of the need to abuse the del.icio.us note field, itself limited to 255 characters (hence the informal machine tag syntax).

There's not a lot of formatting going on, as previously discussed

here <http://www.aaronland.info/weblog/2006/07/03/smoking/#restobook> ,
and

here <http://www.aaronland.info/weblog/2006/07/31/baconmelon/#unfiled>
and

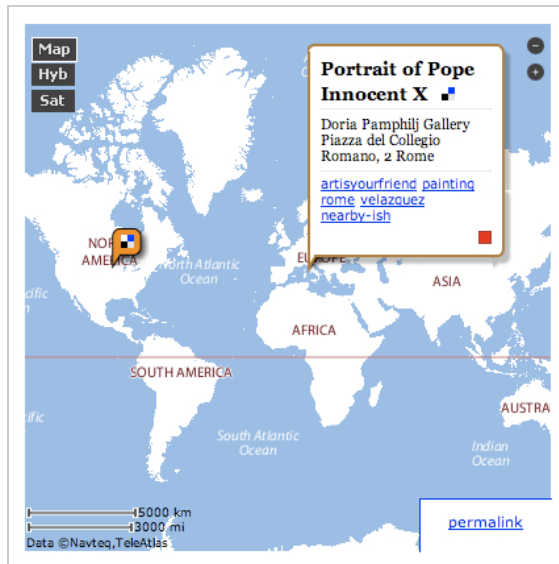
here <http://www.aaronland.info/weblog/2006/08/19/protective/#smallpings> . In fact there's not really a "spec" to format against beyond vague notions of

copying what **my friend Sarah did for her book on cheap places to eat in**

Montréal <http://books.google.com/books?>

[id=lyaqVWzGx5wC&dq=resto+a+go+go+musgrave&printsec=frontcover&source=we](http://books.google.com/books?id=lyaqVWzGx5wC&dq=resto+a+go+go+musgrave&printsec=frontcover&source=we)

b&ots=mZo6SXzvHl&sig=HyY3NZlrS2MzuQsQH6IZ_Q0G4jA#PPA26,M1 . At the moment there are some very basic rules for displaying street addresses and phone numbers and **Yelp** <http://www.yelp.com/> links. I will sort out the rest in time and continue to wonder, some more, about the advisability of **using the syntax to store addresses for, say, museums** <http://del.icio.us/url/0b7384994a0b6360783cb2f8eb4e26d7> .



Anyway : "complex" data in del.icio.us / "pretty" data on the map.

Geotudes

Geowhat? <http://www.geotude.com/>

Yeah, I'm still undecided but figured this was as good a place as any to try them out. Geotudes consist of two parts : A major and minor identifier. Every latitude and longitude can be identified by one of the 65,000 "major" Geotudes representing an area approximately 9,000 square kilometers and an infinite number of "minor" Geotudes. Infinite although by the time you get to 12 points (or 6 pair, each containing two digits and separated by dots) you are dealing with an area

approximately 98 centimeters square. Someone might try to measure England with Geotudes but we'll cross that (tiny) bridge when we get there.



Geotudes are calculated on-the-fly and added automatically (as machine tags) when you save a new location in del.icio.us maps and the major is used to create the **nearby-ish** <http://aaronland.info/www/deliciousmaps/?user=straup&tag=geotude%3A%3A%3D36057> links a place. It's not perfect but everything within the same 80 x 100 (ish) kilometer box is a start and you can more easily fudge things like the San Francisco Bay Area rather than searching for `sanfrancisco + oakland + maybesan jose`. (Which you can't, anyway, but I'll get to that shortly.)

The website states : "*[A] Geotude is permanent and hierarchical. And as a trade-off: Geotude is less intuitive than address, but more intuitive than latitude/longitude. Geotude is more precise than address, but less precise than latitude/longitude*".

Unfortunately, it's not possible to do wild-card tag searches in del.cio.us so there's no way to search for a particular major Geotude and then narrow it down by one or two minor "pairs" (approximately 10 x 10 and 1 x 1 kilometers, respectively).

Still, it's good enough for cities.

And machines.

*If you're wondering, besides writing **Geotude functions in JavaScript** <http://aaronland.info/javascript/geotude/> I also wrote libraries for **PHP** <http://www.aaronland.info/php/geotude/> and **Perl** <http://search.cpan.org/dist/Geo-Geotude> . If someone else would like to write the Python bindings, I would be much obliged.*

Machine Tags

Like the restobook stuff, there isn't too much happening here. Yet. This is still just a point release so the "feature-ness" of machine tags is **they are recognized, and parsed** <http://www.aaronland.info/javascript/machinetag/> , as such. And then mostly not displayed.

Except for Geotudes. <http://aaronland.info/www/deliciousmaps/?user=straup&tag=geotude%3D3D35557>

Barcodes

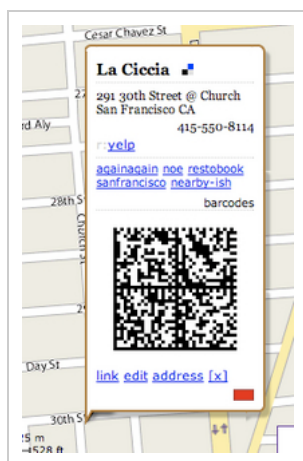
O RLY?

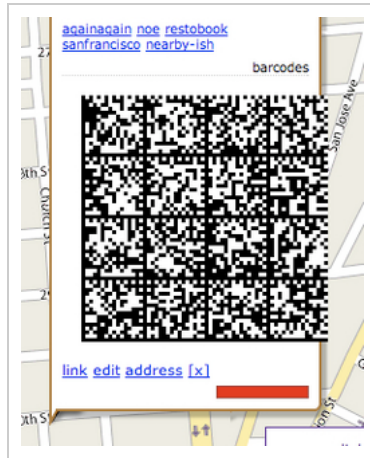
A little silly in the browser, maybe, but they do mostly work and that could be useful for things like phone numbers. Or URLs for — whoa — **mobile websites** http://www.aaronland.info/weblog/2007/07/28/trees/#delmaps_pm .

At the moment I am using **Guido Sohne's SemaFox encoder** <http://sohne.net/projects/semafox> which creates **Semacodes** <http://semacode.org/> . I generally prefer QR codes since they are better suited for arbitrary text, rather than just URLs. There's a **very impressive pure-JavaScript library for generating QR codes** <http://la.ma.la/misc/qrcode/> but it runs ape-shit over all the default `String` and `Array` methods in the language (why do people do that ... it makes

Perl hackers look like prudes, by comparison) so I eventually put it on the back-burner.

My hope was that addresses and phone numbers would be small enough to Just Work™ in a Semacode. This is only Sometimes True™.





Both libraries are, however, pretty slow since they use tables to draw **all those**

boxes <http://www.aaronland.info/weblog/2007/02/17/platform/#barcode> .

Eventually, I may have to write a

Canvas <http://developer.apple.com/documentation/AppleApplications/Reference/SafariJSRef/Classes/Canvas.html> rendering widget for barcodes...

Navigation

Tag-surfing

That is, tags for individual locations are displayed in the marker widget and, when clicked, will redefine the search query and redraw the map accordingly.

That's the good news. The bad news is that you can not poke around the intersection of multiple tags. At the moment the del.icio.us JSON feeds are available only on a per-user basis and limited to a maximum of two combined tags. Since one of the tags has to be `del:bookmark=geo` in order to find stuff with actual geographic coordinates, that only leaves one tag left to play with.

Maybe I will add an optional feature to pull back anything matching a particular set of tags and then loop over them looking for "geo" tags in the browser.

This will produce some weird results for some people but might be a better 80/20 solution for most.

Permalinks

Tag-surfing happens entirely in JavaScript land so that the browser does not need to refresh the page and suck down all the various dependencies. I've also added hooks for the code to read in query parameters in the URL so you can point to a specific tag search or modify it in the location bar.

Ultimately this will need to be expanded to **shamelessly copy the Oakland Crimespotting site** <http://mike.teczno.com/notes/oakland-crime-maps/IX.html> whose permalinks are updated based on the map's position and zoom level.

Icons

Maybe not better icons. But icons.

ph3ar

All the input from del.icio.us is properly sanitized. It's not that I don't trust the del.icio.us kids to send safe data. It's just that not trusting anything that comes back across the wire is More Safe ®.

I say "properly" because I *think* I've covered all the bases but this stuff gets ugly and complicated fast.

Where "complicated" means it's actually all pretty straightforward except for the part where you spend 80% of your time accounting for things like Wikipedia including literal ' ' symbols in their URLs....

If you see something I've missed please direct the clue-bats accordingly.



Going forward, the ballpark "roadmap" in my mind looks something like this

:

- **ongoing** — *Clean up and tighten the design (consider the search interfaces which are, well, bad) and spend some time consolidating the JavaScript code; both of which are starting to look like a neglected squash patch.*
- **0.3** — *Replace the standard Y! Maps marker with a **custom overlay** and ensure that when it is opened and overflows the map container it scrolls in to view* <http://www.flickr.com/map> *. And copy-and-paste which, for some insane reason, is disabled in the standard marker.*
- **0.35** — *Start thinking about compressing all the various JavaScript files.*
- **0.4** — *Pirate maps! And places more precise than just a marker.* <http://adactio.com/journal/1336> *That's fancy-talk for "polylines". This may wait until version 0.5.*
- **0.5** — *Mapstractify* <http://www.mapstraction.com/> *and start*

thinking about how to use available **Open Street**

Maps <http://www.openstreetmap.org/> data, more better. Being able play nice with OSM would go a long way towards making street names in the pirate maps, uh, possible.

- **0.6** — Better reverse-geocoding than whatever comes out of the **Geonames API** <http://www.geonames.org/> . This might be writing something from scratch or just using the **clever driving directions hack** <http://nicogoeminne.googlepages.com/reversegeocode.html> that bubbled up the other day.
- **0.7** — Abstractify the data store layer so that you could use something other than *del.icio.us*. I don't really have any idea what that would look like except to maybe build something **simple using** **Namazu** <http://www.aaronland.info/weblog/2007/06/08/pynchonte/#nmzfilter> and something using **Redland** <http://www.triplr.org/> (or maybe **LARQ** <http://jena.sourceforge.net/ARQ/lucene-arq.html>) for the people who want to do more than "cow-pathing". <http://decafbad.com/blog/2007/08/21/sticky-tags-for-twitter>
- **0.8** — More data, specifically private data, compound tag searches and searches not bound to individual users.

Famous last words, really.

The young architect Charles-Edouard Jeanneret, then working in Paris at the offices of the Perret brothers, witnessed the typical response as his employer, Auguste Perret, burst into the *atelier*, a newspaper in his fist, and shouted: 'Bleriot has crossed the channel! Wars are finished: no more wars are possible! There are no longer any frontiers!' It was not so much that frontiers no longer existed; it was that they were changing, and with them, perceptions and behaviour altered too. In *Aircraft* (1935), Jeanneret, better known as Le Corbussier, remembered the impact a series of historic flights made on Parisians during 1909:

"...from my student's garret on Quai St. Michel I heard a noise which for the first time filled the entire sky of Paris. Until then men had been aware of one voice only from above — bellowing or thundering — the voice of the storm. I craned my neck out of the small window to catch sight of this unknown messenger. The Comte de Lambert, having succeeded in 'taking off' at Juvisy, had descended toward Paris and circled the Eiffel Tower at a height of 300 metres. It was miraculous, it was mad! Our dreams then could become reality, however daring they might be."

— **David Pascoe** <http://books.google.com/books?id=Pp2v1bgmvJEC&dq=airspace&printsec=frontcover&source=web&ots=b6Rndrh8Vp&sig=mT3lqkAi0c7tz7ZCzQ-jNXArJwk#PPA30,M1>

Considering that barcodes were added at the very last minute, as I was working on packaging version 0.2, who knows what I will actually be working on by then. These things seem to **take on a life of their**

own <http://www.aaronland.info/weblog/2007/08/21/address/#doom> which is, honestly, where most of **the magic comes**

from <http://www.canada.com/components/print.aspx?id=deb2350c-8981-4af9-afc0-c0ebbb4b86ae> .



2007-08-22T23:58:14-0700



Things I Am Not Talking About

Talk Is Cheap

Talk Is Cheap

Both proposals I recently submitted to the Talking Head Circuit have been rejected. That's okay, I think.

It smarts, a little, being as vain as the next person but mostly I think it's okay. I was pretty sure that at least one of them would get refused and wrote it anyway as much as an exercise to see how and what I would say. Re-reading them now I'm not sure whether I would change much but then I am also really bad at writing things like artist's statements.

Here for your pleasure and my archival purposes (I have a terrible habit of losing these things) is what I wrote. In both cases, I got a friendly reply to the original submission asking to "expand" a little on just what the hell I was talking about so I've included the follow-ups as well.

Computational Deep-Water and the Gasoline of Internal Desire

The Internet as a single point of failure.

"She has her console and her random access memory. I have my nitwit piece of paper. But I want something to pass between us." -- Don DeLillio

The Internet has rightly been called an
"architectures of participation". Paper, though,

remains the most successful and robust architecture of shared histories to date.

The prevailing theory that all human endeavour will sooner or later migrate to the online sphere ignores the physical constraints that continue to bind the digital world; both as a risk factor and a barrier to its use.

More importantly : We like "things" -- books, the plastic arts, schwag, otherwise cheap souvenirs that become valued artifacts -- because they afford mystery and the room for an object to adapt to the world around them and not the other way around.

If we imagine human language and computers as two equal and opposing forms of magic -- never able to fully understand one another -- then paper can be seen as a bridge, and the papernet as the API, between the two.

"Google has facts. Cities have secrets." -- Kevin Slavin

*Can you dive into some more details with your talk?
Do you know Kevin Slavin?*

This is a totally unedited dump so it may seem a bit scattered but there's nothing there I wouldn't say to someone's face or feel comfortable arguing about.

In many respects the talk is a re-telling of the "Papernet" talk I did last year at XTech. It would be wrong to say otherwise but I think that's okay because there are some interesting ideas in there and I would enjoy presenting them to a larger audience :

<http://www.aaronland.info/talks/papernet/> <http://www.aaronland.info/talks/papernet/>

<http://www.aaronland.info/weblog/2006/12/17/meat/#papernet> <http://www.aaronland.info/weblog/2006/12/17/meat/#papernet>

We've both been around the Interweb long enough to hear people talk about how "paper is dead" and, more recently, that everything is moving in to the cloud. I think this is laughable one because it's not the world I want to live in and two because it doesn't work.

The cloud and the forest (I'm sorry...please kill me now) have different affordances and too little time is spent exploring the relationship between the two.

Take O'Reilly, for instance. I just can't get it to the "Safari" site. We have a commercial license at work but I never use it because it places too many demands on the moment.

That isn't to say that making all that data available online is bad; if there were a proper API so that I (or someone else) could do the functional equivalent of "apropos -k" and then "man" against the Safari dataset I would use it in a heartbeat.

But no one is going to give up the books and if you forced me to read the tea leaves I'd say that's why the company started doing short pieces and distributing them as PDF files. The expectation is that people might refer to the content while they are at their computer but not that they'll actually read it there.

The tools for collaboration and publishing that the Internet have given us are fantastic. But that doesn't mean I want to sit in front of my computer every day.

Plus leaving notes on computers is a real pain, despite some interesting work being done to offline and sync data across sources.

Or news and newspapers :

Everybody (except me) loves portals. And everybody always talks about portalizing all your various new sources and to some extent this is what we've gotten with syndication feeds and news readers.

But I don't really like getting up in the morning and sitting in front of my computer. I do but I also look

forward to Sunday when I can wrap myself in to the Times at home or at Ritual where I don't have to care that they've covered all the power outlets or at the Park where I don't have to care that there are no open wireless networks.

I remember when Salon started their subscription model thinking "That's fine. I would pay for that but I'd like to get a plain vanilla [insert yer data format here] version so that I can print it out and read it in bed or on the bus or whatever."

I know that there are capital-C concerns about leaky content but, frankly, it's going to get out anyway and but shutting everyone else off publishers shut themselves off to the possibilities. I give you the New York Times as Example A :

<http://open.blogs.nytimes.com/2007/10/23/messing-around-with-metadata/> <http://open.blogs.nytimes.com/2007/10/23/messing-around-with-metadata/>

There's also the part where there are no good consumer printers but I'll get to that in a minute.

Or maps. It always comes back to maps, right?

Andrew (Turner) was threatening to make me an interface for actual printable "pirate maps" using the OSM data, at FOSS4G, but I think he's gotten dragged off in other directions.

Here is my shameless pitch to you : Because talk is cheap and I never like presenting without working code, I will make sure that there are tools if you let me speak. I mean I'll do it anyway but... ;-)

And finally, recipes. No one has any idea what to do about recipes because they manage to encompass all the problems in our relationship with computers :

<http://www.aaronland.info/weblog/2007/08/21/address/#doom> <http://www.aaronland.info/weblog/2007/08/21/address/#doom>

From 50,000 feet the talk is about paper. From 25,000 feet the talk is about (some) of the mechanics of bridging the physical and digital world and "content".

But at its core the talk is about the importance of understanding the role and the value of specific media and artifacts in general. Being able to digitize everything and rapidly reproduce stuff is awesome but we are not just a society of bits and there is not always a linear, rationale progression in what we hold to be important.

All data may be ones and zeros but in people's hearts and minds it is not created equal because (cue the social life of documents theme song) given form they are stories and because we can and want to develop relationships with objects.

<http://www.aaronland.info/weblog/2007/02/17/pla>

tform/#wall <http://www.aaronland.info/weblog/2007/02/17/platform/#wall>

Hence the quote from Kevin Slavin. Or something like that. It's all very much a work in progress...

The API as Curator

There is mystery in computers and romance in the Internet despite their formal day-to-day mechanics. Still, few in the increasingly overlapping world of programming and fine arts understand the subtleties of the other very well. Each, in their own way, program only what they know.

"API" is an acronym and computer-speak for application programming interface. An API is like contract dictating the terms by which third-party software (developers) can build their own tools on top of an existing application (API).

More recently APIs - traditionally the domain of desktop or mainframe computers - have migrated to the Internet, typically described as "web services" allowing ideas and services to be cross-pollinated across an even wider arena by an ever-growing number of participants.

The arts community is in a unique position the

explore the Internet and the emergence of communities of collaboration as a medium all its own and to imagine the API as the newest wrinkle in the delicious game of interpretation versus intent.

In the same way that artists embrace their craft, though, the Internet and computer programming need to be better understood on their own terms and treated as more than glorified photocopiers.

Until that happens the best we can hope will be created is more noise in the already deafening echo chamber of misunderstanding.

We're keen to involve you in the ... program, but frankly, your abstract's confused some of the program committee. I wonder if you could help by fleshing out your proposal somewhat.

Many ... involved in the conference are trying social computing in some way and many are mashing up interesting stuff, carrying out activities in Flickr, YouTube, etc., and otherwise exploiting the potential of web 2.0 as well as exploring semantic web, geo-location, and other potentially important approaches to ... So it's important to put your proposal in that context. There are also a group that are quite familiar with 'new media' art -- both creating and curating it.

Could you please push the abstract a bit? We're after pithy, straightforward, and, clear about who you'd like to engage. You should be as technical as you want the paper to be (about APIs for example)

This will help us figure out where in the meeting to slot the paper, and relieve the concerns of the more academically inclined on the Program Committee.

The really blunt version is :

People in the arts community don't know how to program computers and by blinding themselves to

the boring details (mixing your own paint, so to speak) they will never be full participants on the Internet.

Someone invited Sara Diamond to come and speak at Yahoo! about a year ago. It was an interesting talk but, really, nothing I hadn't heard before. A lot of pictures of clothing with embedded computers and waxing poetic about multi-disciplinary projects and bridging the worlds of art and science and commerce.

At the end of all I asked : So, are you teaching any actual computer programming to the students at **OCAD** <http://www.ocad.ca/> yet? The answer was, of course, no.

I still cringe thinking about the many times Alice Mansell told us we were "the visually literate" class (I don't think she actually said "class" for fear of sparking the revolution but you get the idea...) when she was president of **NSCAD** <http://www.nscad.ns.ca/> .

~~She may well have been right but it betrays a depressing habit of peoples in the arts to treat software as just something you throw "code monkeys" at in pretty much the same way that the aristocracy used to throw "crafts people" at church walls.~~

Thinking about it, on the way home tonight, when I said...

"She may well have been right but it betrays a depressing habit of peoples in the arts to treat software as just something you throw "code monkeys" at in pretty much the same way that the aristocracy used to throw "crafts people" at church walls."

...perhaps a better analogy would have been printmaking. Again.

It is true that lots of artists make prints with the help and guidance of master printmakers and you can look at computer-driven works in the same way. But if that's all there was, printmaking wouldn't be very exciting.

I happened to learn lithography from a proper master printer (Bob Rogers) and he would always make snide remarks about the guy who taught printmaking at OCA(D) and encouraged his student to use cracked stones and etch them with Drano and whatever else came to mind.

To say the remarks were snide is a bit mis-leading. They were definitely snide but they were also made with love because Bob also wanted people to feel around for the boundaries of the possibilities. He just thought you'd be better equipped to do that if you actually understood how it (lithography) worked.

As usual, the only ones to mostly get it are the printmakers.

The good kids at Threadless (.com) are printmakers by any measure, even if they don't think of themselves that way, and you could fill an entire day's worth of panels revisiting the arts and crafts debate in the context of Threadless and community and the Interweb.

The photographers are starting to get it. Sort of. Everyone else is still making brochureware or only just discovering Flash animation, ten years too late :

<http://www.nearfield.org/2007/11/making-things-talk> <http://www.nearfield.org/2007/11/making-things-talk>

I don't want to think about the arts as software or, worse, services but we have built this beautiful monster, called the Internet, that allows people to explore and share in ways we've never seen before.

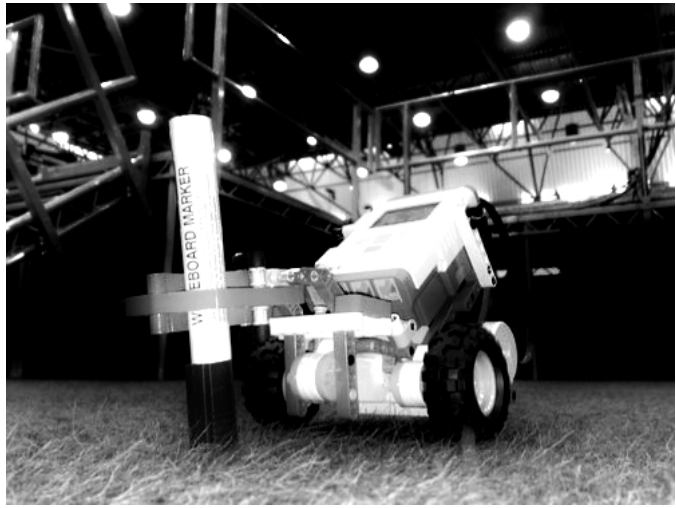
Where the advent of photography meant learning and understanding the vagueries of the darkroom the Internet -- and programming and APIs -- deserves a measure of attention that remains unmet mostly out of fear and outdated prejudice.

That's what I want to talk about. At a brass tacks level, it means demonstrating the value of providing real funding in both schools and institution to teach the skills and then to let people poke around and find the magic.

Lost in Transanimation

Maybe it all just needs a little more time to gestate but,
while that's happening, *I'd go see those*
talks [http://www.aaronland.info/weblog/2008/04/23/things/
#things](http://www.aaronland.info/weblog/2008/04/23/things/#things) .

2007-12-12T08:04:00-0800



Little Partial(ly Cloudy) Castles

Release notes are the new...

old(1.0) != new(1.0)

My Inbox is full of rocks

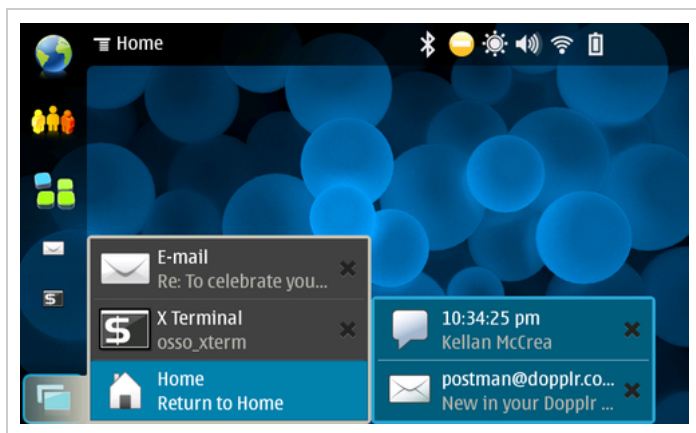
Your door is an Interjar

Release notes are the new novel

It goes without saying that (nearly) everyone brought their laptops to **the News Years weekend in**

Sonoma <http://flickr.com/photos/straup/sets/72157603617830727/> .

I played a game of semantic bullshit and brought only **a fancy-pants Nokia tablet** <http://flickr.com/photos/straup/tags/sketch/> . One of the things I like the most about the new "OS2008" models is the polite throbber that gently blinks when you receive email, or an instant message or some other event that an application thinks it should tell you about.



You can see **where this is** **going** <http://www.aaronland.info/weblog/2007/12/20/castles/#growler10> .

There are even **Python bindings for sending various types of notifications** http://maemo.org/development/documentation/apis/3-x/python-maemo-3.x/python_libosso_api.html but since I still haven't figured out the magic to talk to the throbber I spent the weekend periodically poking at the **plain old web** <http://del.icio.us/straup/plainoldwww> to see what was going on.

Of course, when I got home **all the little Growl-based notification agents**

I've been

writing <http://www.aaronland.info/weblog/2007/12/20/castles/#zomg>

woke up and spewed four days worth of hot friend action all over my screen.

Computers. **They're dumb like**

that. http://radar.oreilly.com/archives/2008/01/human_vs_machine_google_wallstreet.html

So, now you can specify a `skip_items_older_than` preference in the config file and ignore anything older than an hour or a day or whatever. I suppose it might be neat to write some magic to set that option dynamically based on the last time an application was able to register the presence of your `pho^H^H^Hdevice`, but not today.

Instead, there is only the quick and dirty :

- ***growler.py*** <http://aaronland.info/python/growler/growler-1.1.tar.gz> ***1.1***
- ***flgrowler.py*** <http://aaronland.info/python/flgrowler/flgrowler-1.1.tar.gz> ***1.1***
- ***twgrowler.py*** <http://aaronland.info/python/twgrowler/twgrowler-1.1.tar.gz> ***1.1***

Meanwhile, here's **a long photo of us eating dinner** <http://www.vimeo.com/466557> .

old(1.0) != new(1.0)

```
118 | <ol>
119 |   <li><em>flactivity</em>, to display recent activity on your photos</li>
120 |   <li><em>flcontacts</em>, to display recent photos from your contacts</li>
121 |   <li><em>g</em>
122 |     <li><strong><a href="">twgrowler.py</a></strong>, </li>
123 |   </ul>
124 |
125 |   <p>The only caveat is a known bug where searching for nearby
126 |   Bluetooth devices will get stuck in some sort of infinite
127 |   loop. I don't know why or, even, how to reproduce the
128 |   problem. From a distance it appears to be something
129 |   happening at the (OS X) operating system layer but more
130 |   investigating is necessary.</p>
131 |
132 |   </ins>
133 |   </ins>
134 |
135 |   <ins class="post" id="rocks" datetime="2007-12-26T10:27:52-0800">
136 |     <ins class="content" datetime="2007-12-26T10:27:55-0800">
137 |
138 |       <h1 class="aaronland email intercloud perl visualization whosonfir
139 |
140 |       <p>This morning, I was reminded of the
141 |       quick-and-dirty-and-ugly <a href="http://www.aaronland.info/weblog
142 |       the relationships between email messages</a> that I wrote a
143 |
144 |       <p>(nXML Invalid CVS:1.43 Fill)--L121-- 5%-----
145 |       castles/index.xml
```

The code discussed in this

post <http://www.aaronland.info/weblog/2007/12/20/castles/#zomg> has been pulled apart and packaged in handy little feature-specific bundles complete with proper Python installation scripts. They are :

- **growler.py** <http://aaronland.info/python/growler/> , a base class for creating Python daemons to poll services and displaying updates with Growl notifications
- **flgrowler.py** <http://aaronland.info/python/flgrowler/> , a base class and three separate applications :
 1. *flactivity*, to display recent activity on your photos
 2. *flcontacts*, to display recent photos from your contacts
 3. *flavourites*, to display photos recently favoured by your

contacts

- ***twgrowler.py*** <http://aaronland.info/python/twgrowler/> , a base class and one application :

1. twgrowler, to display recent Twitter messages from your contacts

The only caveat is a known bug where searching for nearby Bluetooth devices will get stuck in some sort of infinite loop. I don't know why or, even, how to reproduce the problem. From a distance it appears to be something happening at the (OS X) operating system layer but more investigating is necessary.

As mentioned, there is nothing especially OS X specific about any of this beyond the Growl bindings and every operating system has at least one functional equivalent. At some point, I may also port it to use any number of notification agents.

But not today.

2007-12-29T12:23:29-0800

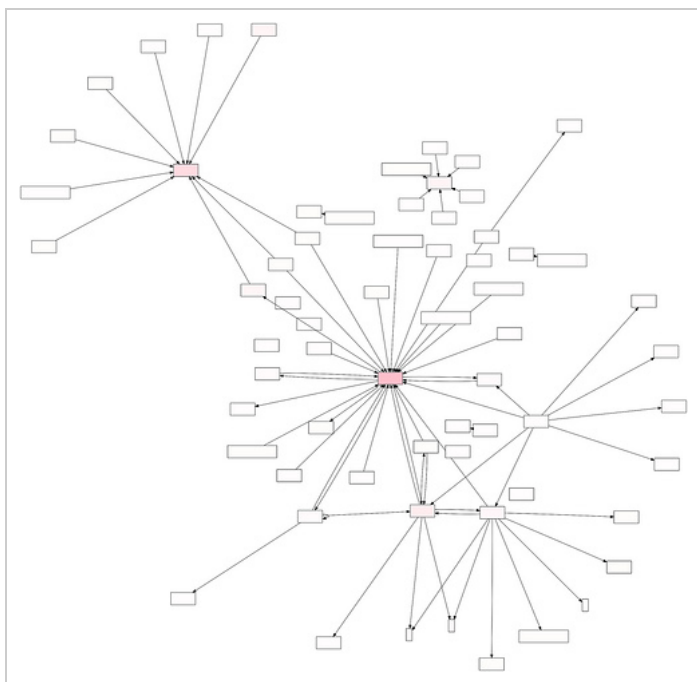
My Inbox is full of rocks

This morning, I was reminded of the quick-and-dirty-and-ugly **Perl module for graphing the relationships between email messages** <http://www.aaronland.info/weblog/2004/06/07/5527/> that I wrote a few years ago. As usual there is no magic since it simply hands all the hard work to **GraphViz** <http://search.cpan.org/dist/GraphViz/> and treats everything as **sticks and rocks** <http://www.tom-carden.co.uk/2007/11/08/sticks-and-rocks-illustrating-the-problem/> .

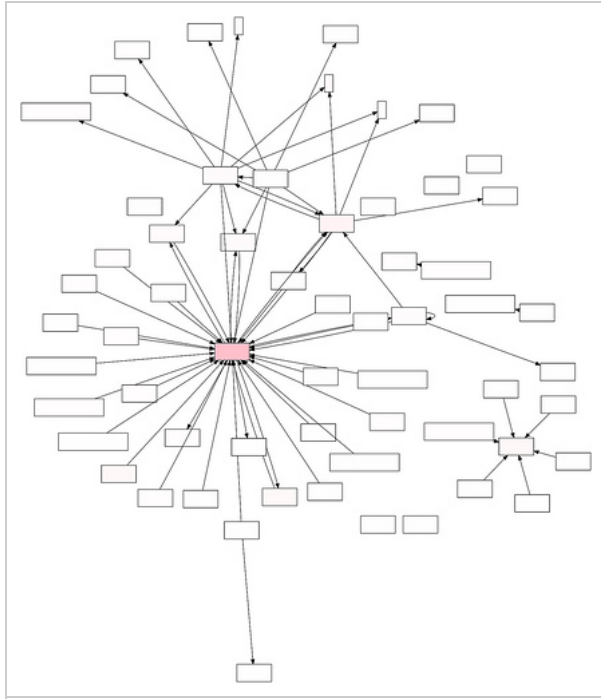
I know that some people like to imagine that there is actual knowledge (or if you are a timesome doorknob, "agency") to be teased out these things but I prefer to think of them more as **Auggie Wren** <http://www.npr.org/templates/story/story.php?storyId=4244994> devices. The value for me is to collect a lot of them and then **try to feel the elephant** <http://www.aaronland.info/weblog/2007/12/20/castles/#zomg> over time; to imagine the patterns as **a halftone screen with which to rasterbate** <http://flickr.com/map?q=rasterbate> everything else.

With that in mind I dug out the code just to see if it still worked and, in the process of all the usual bugfixes, decided to teach it a few new tricks.

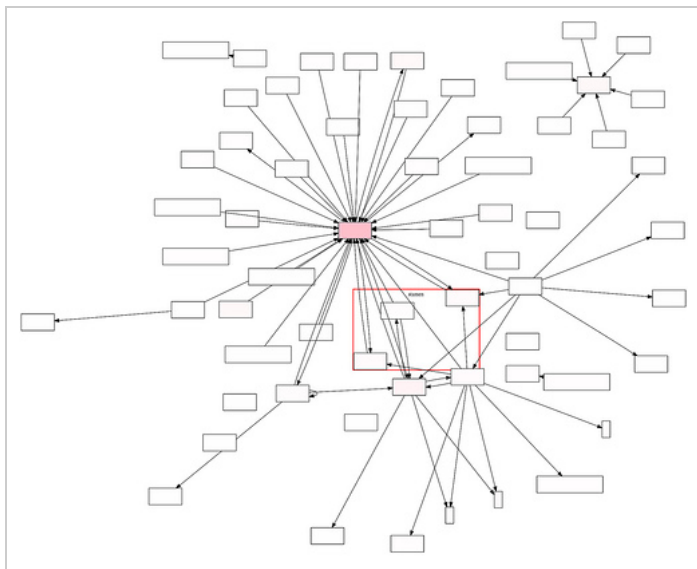
Here's my Inbox, with just basic **To :** , **From :** and **Cc :** relationships plotted :



Here's the same thing with three sets of email addresses treated as equivalent and aliased to each other when the whole thing is rendered :



Here's the same thing with both equivalencies and a "cluster" of email addresses — for example, all the addresses for a group of people who work together — represented by ... a box. *Oh well* :



Finally, the code <http://aaronland.info/perl/ascope/email/graph/> to do all of this is :

```
my $graph = ASCOPE::Email::Graph->new({'obscure' => 1});
$graph->add_mbox("/path/to/mbox");

$graph->add_equivalencies(["me@work.net", "me@home.net",
                          "me@gmail.com"]);

$graph->add_equivalencies(["buddy@cheezburger.com",
                          "bud.dy@ohhai.com"]);

$graph->add_equivalencies(["girly@cheezburger.com",
                          "grrrrr.ly@ohrly.com"]);

$graph->add_cluster(["buddy@cheezburger.com", "bud.dy@ohhai.com",
                    "girly@cheezburger.com", "grrrrr.ly@ohrly.com"],
                    "cheezburger");

$graph->as_png("/path/to/graph.png");
```

Clearly the next thing to do is to teach the code to read in an **address book** <http://www.aaronland.info/weblog/2007/08/24/aware/#reduced> and generate those equivalencies and clusters on the fly. Maybe I will do that before another three and half years have passed. Colour-coded arrows too but that looks like it might require actual work...

Related, there are some interesting threads to pull by **FOAF-**

ifying <http://search.cpan.org/dist/XML-Generator-RFC822-RDF/> all those addresses and **feeding them back in to the**
cloud. <http://laughingmeme.org/2007/12/14/amazoncom-simplydb/> (The code already supports the hashing of addresses; I just obscured everything in the examples above because I ph33r you.) I doubt this will ever happen because one side-effect of the current **social share-my-graph**
fetish <http://www.mediamatic.net/article-26386-en.html> is that it will be even easier to dereference and harvest mbox_sha1sum addresses — and the Creepy People surely will — but there are certainly some possibilities to explore.

Meantime : **ASCOPE::Email::Graph.pm**

0.2 <http://aaronland.info/perl/ascope/email/graph/ASCOPE-Email-Graph-0.2.tar.gz>

2007-12-26T10:27:55-0800

Your door is an Interjar

With the enormous amount of money and effort it takes to get together with people you barely see, all of the tensions will be exaggerated. I can see people saying, "I spent all this money, I came all this way and for what?" It's supposed to be a certain way. It puts an enormous amount of pressure on those kinds of contacts and connections that wouldn't be there normally.

Vered Amit <http://www.nytimes.com/2007/12/23/weekinreview/23kershaw.html>

ZOMG!! ADDRESSBUCKET RUN OUT OF INODEZ!!!!

Environment `http://nextbutton.pureprofile.com/TVC/?id=4326` "

eht erehw si em skri taht trap ylno eht ,edisa (laintnegnat ylnisaercni) na sA
 -ngised dradnats si sihT .xednaps ."yalp dna semag" eht pu depparw si aedi elohw
 uoy ekam taht snottub emas eht lla sehsup dna ,eb dluohs ti llew sa ,emahs-rof
 ekil I .sroolf eht pom ot flesruoy ecrof ot redro ni rennid rof revo elpoep etivni
 hcturc elbatpecca ylctefrep a si siht tub (eil a s'taht ,yllautca) oot semag gniyalp
 eb tsael ta s'tel os deerg dna htols namuh fo eussi regral eht no krow ew elihw
 .snoitavitom ruo tuoba tsenoh



But that's not what I wanted to talk about.

At one point he says "Well, these are just mock-ups. We probably don't want to walk around with big numbers floating over everything. We'll want something more in the *design of everyday things* vein."

Which I thought was curious because it seems so very wrong. Or rather, no one really wants to *interact* with any of this data. They just want it to be **continuously partially present** <http://www.aaronland.info/weblog/2006/06/12/areacode/#ole> .

Consider Twitter : The part where your phone is either pestering you its horrible YOU HAVE A NEW TEXT MESSAGE sound every two or three minutes, just so someone can tell you they're making hummus — or the part where Continuous Partial Attention is greedspeak for Death By Micropayments if you don't have a phone with unlimited text messages — generally means that people

turn off the notify-by-SMS feature.

I would have never bothered with Twitter if I hadn't able to have incoming notifications sent to my IM client where I can continuously partially ignore them.

ZOMG!!! WE IS TEH ROBOT SOCIALLORDZ IN UR SAVIN!!!!

But when I woke up with a hangover, last week, I couldn't even be bothered to reach out and switch windows, and then fiddle with tabs, just to know that

Dunstan was doing push-ups in

bed <http://twitter.com/dunstan/statuses/497682572> . I did not want to move but I was awake and staring vaguely at my screen hoping that the power of my social network would simply give me **Digital Brain**

Rub <http://incubator.quasimondo.com/flash/flickeur.php> .

Having nothing at all to do with hangovers, you can trace this long line of desire from the movies, to television, to IRC backchannels and

hecklebots http://joi.ito.com/archives/2003/07/25/hecklebot_arrives_and_appears_in_new_york_times.html at conferences, to Twitter itself and then the

public Twitter display <http://flickr.com/search/?q=twitter+sxsw&ss=2> at SXSW in 2007 (hardly novel, O'Reilly also did something similar with its "attention

stream" <http://www.flickr.com/photos/seanbonner/6630754/in/pool-etech/> at ETech in 2005), to screensavers, to digital frames ranging from

consumer level devices tied to partner providers to **spaceship-aware Linux distributions wrapped up in shiny paper** <http://www.theredpost.com/kit/>

all the way to full-on **Ray Bradbury**

installations <http://www.albersfoundation.org/> .



I know some people who say things like : "I want to be permanently jacked-in to the Internet! I want to have a direct IRC-link to my brain!! I WANT TO SEE LOGFILEZ ON UR FACE!!!!" Okay, no one's actually said that last part but it's what they're thinking. And while I think it's mostly crazy talk — try to imagine the world around you pulling a **cover flow** <http://swik.net/coverflow> everytime you turned your head — I also think it ultimately comes from a good place.

One really fantastic by-product of the Network is that it has afforded people, still separated by the cold hard heart of flesh and blood and time and space, the ability to maintain enough shared experience to keep relationships in bloom. Make no mistake : We all die some day and no one wants to be lonely in the tiny bookends we manage to tease out of time's ... continuous partial attention.

The argument that Internet is, as the drunken 23-year old PolySci student I foolishly tried to have a rationale argument with described, a "great isolator" has some merit. Watching people pass hours in front of their computers playing games, writing code or ranting (is there a difference?) otherwise indifferent to any warm-blooded creature nearby is enough to give anyone pause.



I've watched the cycle enough, though, to be pretty certain that's not the case. Sooner or later all communities that comes together online meet offline whether it's gamers who fly to distant towns to, yes, play games *together in the same room*, or Flickr users who meet to take pictures together, to the **2 to 70 people who have met regularly every month** <http://www.yulblog.org/> for *six years running* and share nothing more in common than a city and an appreciation of weblogs, the Internet and beer.

I think everybody instinctively knows that a life online is not enough by itself but that understanding is also not always enough to push us beyond the prejudices and insecurities we bring to a room full of strangers. When it works, the Internet and an individual's ability to project something about themselves "out there" provides just enough of a bridge between people who've never met to want to entertain the idea of a friendship.

"Accelerated

serendipity. <http://www.canada.com/montrealgazette/news/business/story.html?id=de5d3f2b-7aae-42ee-8230-a70c92a8a753&k=44586> "

Societies have always had norms and conventions for doing the same thing and they've always kept pace or adapted themselves to the available technologies. That the Internet allows people to More Better dance the tune on their own terms I think is an improvement.



But I was talking about Twitter, wasn't I?

ZOMG!! I WAS YAWNING AND THEN I BURPED!!!!

If the Internet is a single bridge connecting two people at the start of a relationship I like to imagine it becoming **a series of bridges** http://thislife.org/Radio_Episode.aspx?episode=345 over a canal that keep the land on either side of the water from straying too far.

Again, you could replace "Internet" with "available media" and follow the same thread back through history as far as you wanted.



What's changed obviously is the collapsing of the space between the sending out and receiving of our bird calls. A permanent arrivals and departures lounge of human relationship, where everyone you know is paging you on the white courtesy phone (in your pocket).

a gnimoceb won ylno s'ti ecnis tub tuoba deklat neeb gnol sah oot sihT

*ton ro rehtehw tuo erugif ot gnilggurts yllanif er'ew os ytilaer yad-ot-yad lacitcarp
.aedi doog a saw ti*

Chris summed it up

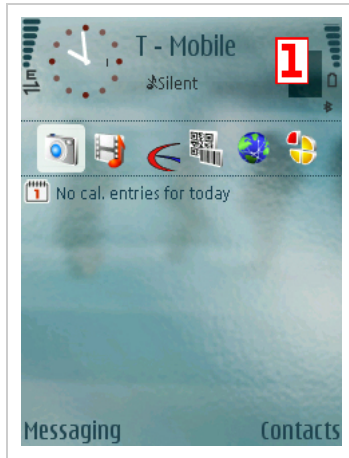
nicely <http://antimega.textdriven.com/antimega/2007/03/28/service-design-notes-increase-serendipity> earlier this year : "Of course, it doesn't mean anything will necessarily happen, but you've created a possibility space for things to happen in."

I sometimes complain about especially enthusiastic Twitter contacts but, really, I am delighted by the whole thing. I love being able to share a frame of reference — a pithy text messages or photos or music; anything important ("important") enough that you might share it over a meal, — with people who can't be near, either because of geography or mundane daily circumstances. It's a kind of jerky stop-motion animation frame of reference, for sure, but without it you are deaf to the tiny anecdotes in a person's life; the daily rituals and gradual changes in tone and focus that help to nurture the understanding and appreciation of a friendship.

A "possibility space".

ZOMG! IM IN UR MEMREEZ PRETENDIN I WUZ THERE!!!!

Of course, it doesn't take long before it becomes a lot of, and usually too much, work to sift through all that chatter. Especially when you're hungover. Which is why I sitting there, drinking coffee, thinking : Wouldn't it be really nice if new messages could Just Appear in tasteful, discrete windows. Something like Growl notifications. Then the nice people at Google reminded me that I had **already written** <http://www.aaronland.info/weblog/2007/01/03/itemz/#disco> the thing that I was looking for.



At the time, I still wanted to get notifications on my phone but found everything about the experience of receiving them via SMS unpleasant. Instead of my phone acting like a hyper-active "personal assistant" squealing with delight every time a new message arrived, I wanted something more like ... a mood ring.

It would be silent, except for maybe the option to vibrate and instead of conventional system notifications it would just hover a great big number over the "idle" screen indicating the number of unread messages. Okay, that's a half-truth if there ever was one : I did add the big floating numbers but that was as much to act as an indicator that the phone's operating system hadn't silently terminated my application.

Since then I've grown fond of the big numbers on their own merits. What I'd really like is to *fill* the idle screen — when the phone is, uh, idle — with those big numbers.

Possibly ambient.

I wrote a Growl enabled version, using the Twitter API, mostly as an afterthought because it seemed like a neat trick and it was an additional 4-5 lines of code.

I never got very far with it. The actual interface for reading the messages was crude at best and I had spent most of my time thinking about and then building the floating numbers. It was also during a period when the Twitter API itself was pretty flakey so the messages I did receive always seemed strangely out of sync with reality. It was also around the time the **mobile version of the site** <http://m.twitter.com/> went live and, experience porn aside, that ultimately proved less demanding to use.

As for the desktop version, it suffered from the same problem that plagues feed aggregators : keeping track of what has or hasn't been seen. Not a hard problem but it's an extra bit of complexity (remember, I was also trying to write an underlying layer of code that would continue to work on my phone...) that makes you wonder whether you couldn't be doing something more fun with your time. I also tend to work on a variety of different computers and the nuisance of doing the magic sync dance between them, so that I don't always see the same 25 notifications everytime I log in to a different computer, finally answered the question for me.

ZOMG! IM IN UR CLOUD CRYIN UR TEARZ!!!!

Did I mention that I was hungover and I was smitten, once again, with the idea of little, well-behaved messages hovering on the periphery?

I often work in the kitchen when I'm at home and what what I've wanted for a long time was a sexy hecklebot, hanging over the counter, that could talk to all the various APIs out there and "show" me what's going on in a world that consists only of my friends.

ekil secivres tahw si siht taht ekam ot tnemugra elbanosaer a s'erehT

.ton s'ti koobecaF etah I ecnis tub od koobecaF

This has always been my attraction to the **RedPost/Kit digital frame** <http://www.theredpost.com/kit/> which does something like this already for Flickr feeds. Maybe still but I wasn't about to get one that afternoon so I continued to sit and stare at my laptop.

There was no blinding flash of insight or anything.

I had already resigned myself to the constraints that Growl imposes on its notification windows : icons are a fixed size; there is no (easy) way to define custom callbacks when a message window is clicked and no way to render or format messages as anything but plain text. The only issue was how to deal with keeping track of messages across multiple computers. I had decided to build myself a notification system for Flickr events.

Or, in the English people who aren't venture capitalists use, "I don't kill my own chicken for Sunday dinner anymore, why should I provision my own server?"

Jacob Harris <http://open.blogs.nytimes.com/2007/12/21/a-first-look-at-amazon-simpliedb/>

I don't actually agree with the underlying argument that's being made there but it does nicely sum up the part where in order to do anything across multiple computers you need something that looks like a database (even if it's just a list of unique identifiers) and then some sort of web interface and then a security model and then all the other crushingly boring details just so you can say : $26452625141 = 1$.

It's possible that my reluctance to use any of the web services (AWS) that Amazon has been releasing over the years has been motivated more by ph33r rather than reason but I've never completely understood *what or why* they were getting at by offering any of this stuff so I've kept to the sidelines and watched.

I have already used

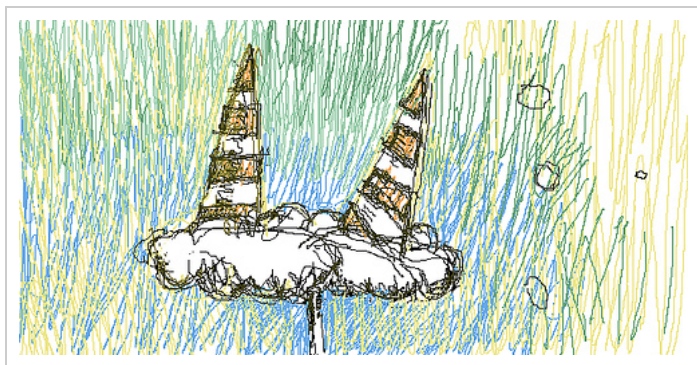
del.icio.us <http://www.aaronland.info/weblog/2006/07/03/smoking/#restobok> and

Stikkitt http://www.aaronland.info/weblog/2007/07/28/trees/#s60_stikkittapp as **floating**

databases http://www.aaronland.info/weblog/2007/08/24/aware/#delmaps_0

2 for other projects but using either to store nothing more than a boolean value or a timestamp seemed about as classy as asking Flickr to "store text files".

Even the still nascent **DWIMD^H^H^HSimpleDB** <http://www.aaronland.info/weblog/2007/08/24/aware/#mtdb> (which will probably do more to advance the ideas behind the Semantic Web than all the articles TBL could write on the subject in a lifetime) is overkill, though it might be sort of interesting to go back and build a day-by-day scrapbook of notifications.



But I digress. I decided to bite the bullet and try **S3** http://en.wikipedia.org/wiki/Amazon_S3 .

ZOMG!!! MOUTHFACE FULL OF WORDZ!!!!

Instead of reading or writing a flag to a local database, the code fetches or creates a web page whose URL contains that flag. Pedants will tell you that is to : GET or POST a "resource" on the Internet; which is also why they're a bunch of tiresome doorknobs.

This has the advantage of being both portable and fantastically easy as well as flexible since the thing at the end of the URL is, ultimately, just a text file that we can append stuff to ad nauseam (or up to 5GB if you are a tiresome doorknob).

And is costing me both cash dollars and some amount of privacy.

S3 is a billable service and Amazon is a big centralized for-profit company with a penchant for data-mining. I have yet to get my first bill but between local caching, the relatively tiny rates of data transfer I am doing and the quick-look-over-here truly micro-payments schedule that Amazon has put in place I figure it's worth trying until my next credit card bill arrives.

The privacy thing is stickier and I remain skeptical about putting anything I really care about in Amazon's cloud. In principle, the things I'm looking at on Flickr could come back to haunt me but my gut tells me that there's nothing I wouldn't freely admit to and if I really feel like that's a plausible risk, today, we're already fucked. Plus I am hashing the keys for every event notification.

Then I remembered that this entire exercise was about active notifications and not building an answering machine in the cloud :

```
$> python ./bin/flickr/flgrowler/flcontacts.py -c flgrowler.cfg
[Thu Dec 20 21:44:58 2007][flcontacts] sleep for 120 seconds
[Thu Dec 20 21:47:13 2007][flcontacts] can not find beaconing device, skipping
[Thu Dec 20 21:47:13 2007][flcontacts] sleep for 120 seconds
[Thu Dec 20 21:49:28 2007][flcontacts] 0a:18:42:b8:30:4a is nearby
[Thu Dec 20 21:49:40 2007][flcontacts] received 17 new events (50 total)
[Thu Dec 20 21:49:58 2007][flcontacts] sleep for 120 seconds
```

(If blog posts had code-names (we would be doomed) then this one would be : To Timo, with love <http://www.nearfield.org/> .)

I have given the "near-field communication" people a hard time over the years mostly because some of the **creepier privacy and security issues** <http://www.thenation.com/doc/20040216/garfinkel> are waved away with Dubya-like assurances of "technological advances" that do not inspire confidence, but even I can see the potential. I just think — hope — that after it's baked for a while it will be more ... yes, serendipitous in nature than "utilitarian" or a tool for "discovering" the **nearest fucking Starbucks** <http://aaronland.info/talks/sxsw07/> .

*,remmus siht nodnoL ni ,tneve yaD kcaH KU eht ta stcejorp eht fo elpuoc A
ot pu tog yeht nehW tub krowten htooteulB eht htiw cigam gnimrofreP tliub erew*

*saw erawtfos eht taht ecneidua eht ni secived htooteulB ynam os erew ereht omed
..seitilibissop eht fo thgiew eht rednu dehsurc*

Anyway, if two machines are running event listeners simultaneously — each reading and writing to the S3 cloud — and you don't happen to be in front of the one that receives an event first you'll never see it since by the time the second computer finds out about it it will have been marked as "seen".

So, accepting the truism that people generally keep their phon... I mean, their devices nearby (enough), this is how we handle the problem :

*1. **Start the application** or wake up from a previous cycle.*

*2. **If the config file** has a list of Bluetooth devices :*

- *First check to see we have the address of a previously seen device and explicitly request the device name*
- *If not, or the operation fails, perform a general query of devices nearby, and look for a match with the list defined in the config file.*
- *If present, store it in memory so that we don't (necessarily) need to do a broad poll on the next pass. This may just be a function of my not understanding how the PyObjC Bluetooth bindings work but in casual testing a broad "find devices" query would often time out without necessarily getting to a known-good device. So this is just easier.*
- *Also, I know about and love the **Lightblue Python Bluetooth bindings** <http://lightblue.sourceforge.net/doc/index.html#finddevices> but if you call the "finddevices" method a second time before the first instance has completed it will make your CPU cry. Talking to the*

Bluetooth frameworks directly is easier which should tell you something given how bent programming in Objective C is...

3. Open the datastore.

- *If the config defines the datastore as an AWS S3 bucket then the application will reach out across the network and ask AWS for a list of all the buckets associated with the access key defined in the config file. If the bucket described in the "s3_bucket_prefix" config is not present it will be created.*

4. Fetch and parse a list of "events" via the Flickr API. *Events is just fancy-talk for a list of results for a question you've asked Flickr : What sort of activity has there been on my photostream? What photos has this user favoured? And so on*

5. For each item *returned generate a unique identifier and check to see if there is a corresponding key in the datastore. If there is, the item is considered as having already been seen and is skipped.*

- *If the config defines the datastore as an AWS S3 then the application will reach out across the network to see if there is an actual "thing" on the Internet that can be addressed with the UID.*

6. For any remaining items, *the application will build a custom message, fetch a thumbnail for the corresponding photo and issue a notification request to the Grawl daemon.*

- *At the moment thumbnails are stored in what your operating system thinks is its "temporary" directory. There is currently no attempt to clean up old images. That, along with a user-defined directory in which to store thumbnails, will be added in future releases.*

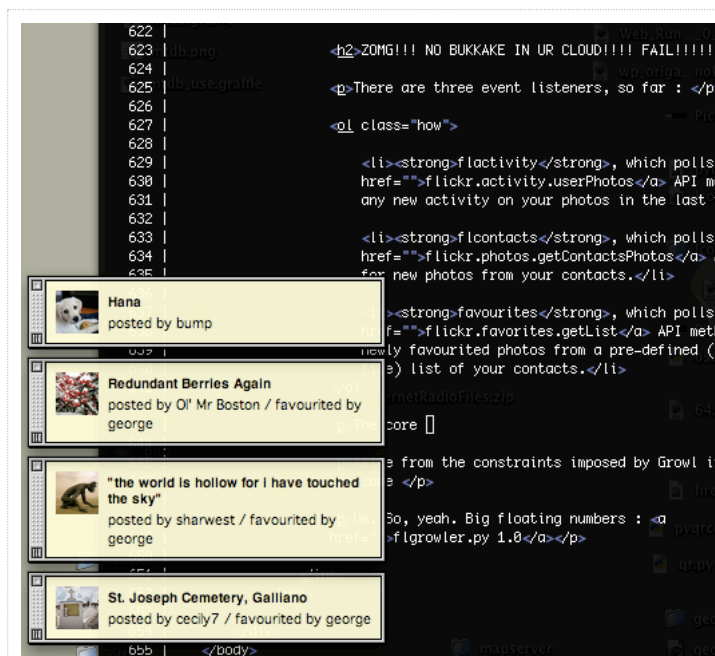
7. **Add an entry** for the unique identifier in the datastore. If the identifier is the key in the datastore then the value is the current time formatted as a Unix timestamp.

- If the config defines the datastore as an AWS S3 then the application will reach out across the network and create a "resource" on the Internet that can be addressed with the UID.

8. **Sleep** for (n) seconds as defined in the application's config file.

In the end it looks something like this :

ZOMG!!! NO BUKKAKE IN UR CLOUD!!!! FAIL!!!!!!



<http://flickr.com/photos/george/favorites>

There are three event listeners, so far :

1. *flactivity*, which polls the **flickr.activity.userPhotos** <http://www.flickr.com/services/api/flickr.activity.userPhotos.html> API method for any new activity on your photos in the last two days.
2. *flcontacts*, which polls the **flickr.photos.getContactsPhotos** <http://www.flickr.com/services/api/flickr.photos.getContactsPhotos.html> API methods for new photos from your contacts.
3. *flavourites*, which polls the **flickr.favorites.getList** <http://www.flickr.com/services/api/flickr.favorites.getList.html> API method for newly favoured photos from a pre-defined (in a config file) list of your contacts.

The core event listener functionality will shortly be packaged as a separate library so that I can write a

Dopplr <http://www.hackdiary.com/archives/000104.html> agent, finally rewrite the Twitter version and then whatever else comes to mind.

Aside from the limits in functionality imposed by Growl itself there are two immediate concerns I'd like to address next :

The first is what to do about window spam after you've been away from your computer for a while. When I sit down at my laptop in the morning the first pass of the newly active listeners will tile my entire screen with event windows. I have taken to scanning them quickly and then just restarting Growl since clicking on each window, to close it, is really boring. Perhaps, the easiest thing to do would be to create a digest of all activity older than, say, an hour and display it as a single notification...

The second issue won't be an problem until I work up the courage to port all of this to Series60 Python but when I do the NFC beaconing code will need to be rewritten to check not only that a given device is within range to display events but also to *stop* displaying events if another device is within range. That way, you could

receive notifications on your phone while you're out and about and then have them switch your computer when you get home.

Or, more likely, have notifications automatically disabled whenever your boss gets within 32 feet...

ZOMG!!! I R NOT LOLCOLORZ!!! BUCKET FULL OF FREEDOM!!!!

Finally. I still find the whole "river of dashboard" metaphor, in which I would include the Growl-y bits I've been working on, mostly boring.

I use a feed aggregator for anything that could loosely be classified as news and lifestyle porn and it works great. I even use it to find out when people I actually care about have posted something to their weblog but I absolutely hate it for *reading* what they've written. Maybe I am the hold-out but I still value the web for the time and effort and consideration that a person puts in to their web *pages* and I think that gives whatever they are talking about context.

Leave It With the (Virtual) Doorman



Operators at Cyberdoorman in the Bronx monitor multiple buildings at once.

Marilynn K. Yee/The N

<http://www.nytimes.com/2007/12/23/realestate/23cov.html>

If I want to be notified at all of when something new has happened, I would like a way to feel the (proverbial) elephant rather than have it join me at the table. It's a bit vague, obviously, but there's got to be something more interesting — more fun — than the way we recycle everything as slices of 99 cent pizza, today.

You'd be forgiven if your bullshit detectors are ringing right now because it's definitely a bit pie-in-the-sky. What little I am sure of is that I don't enjoy the way we leave markers in the forest of songlines that the Internet has become and that there's room for **a little more**

art <http://blog.dopplr.com/index.php/2007/10/23/in-rainbows/> in it all (and maybe an overdue appreciation of **Josef**

Albers <http://www.albersfoundation.org/>) because that is what relationships are made of.

So, yeah. Big floating number friends : **flgrowler.py**

1.0 <http://www.aaronland.info/flickr/flgrowler/> and

beyond <http://www.aaronland.info/weblog/2007/12/20/castles/#growler10> .

ZOMG!! IM IN UR INTENT RUNNIN FROM UR INTERPRETIN!!!!

Update.

Not long after this piece was first posted Adam Greenfield found it in his referer logs and wrote a frank and biting rejoinder. I mention it, here, because :

1. *Questions about my biases aside, he makes a compelling argument and has a point.*
2. *I wouldn't change anything I wrote. Or, put another way :*

From : Aaron Straup Cope

To : Adam Greenfield

Subject : point taken

If there was intent, vis-a-vis the ratio of male to female pointers, in that post then it certainly wasn't conscious.

As for the bit with the bukkake, there's really nothing to read into that. Really. Bad taste? Perhaps. I can live with that.

If someone wants to read anything more than me making a silly joke about the underwhelming nature of the post -- which was really a ridiculous amount of background leading up to some pretty Spartan code -- then I'll meet them at the bar because we'll have an argument to work through.

But, point taken.

Cheers,

The only part that I *will* take exception to is the pithy comment about the "del.icio.us-style" links at the top of these pages.

They are, in fact, a collection of links that **the people in my del.icio.us network** <http://del.icio.us/network/straup> have posted, pulled automatically from **the page's RSS feed** <http://del.icio.us/rss/network/straup>, out of which a new list containing the most recent post per contact is generated. If that list happened to contain a perceiv^e_i[^]_H[^]_Hambient gender bias on the day that Adam came by, then ... I guess I blame Joshua.

I expect that we'll see more misunderstandings like this, where meaningful intent is confused with the automated faceting of a person's presence on the Network, going forward.



We found **1 result** for photos matching **Interpretation** and **versus** and **Intent**.

View: [Most relevant](#) • [Most recent](#) • [Most interesting](#)

Show: [Details](#) • [Thumbnails](#)



Interpretation versus Intent

Uploaded on [6 August 2006](#)



By **straup**

See [more photos](#), or visit [straup's profile](#).

[cameraphone](#), [sanfrancisco](#), [filtr](#), [geonames:locali](#)

Anyway, if you've made it this far **go read what Adam has to say.** <http://speedbird.wordpress.com/2007/12/26/welcome-to-the-desert-of-the-male/>

2007-12-16T20:02:00-0800